

Oracle® Universal Installer

Concepts Guide

10g Release 1 (10.1)

Part No. B12140-01

December 2003

Oracle Universal Installer Concepts Guide, 10g Release 1 (10.1)

Part No. B12140-01

Copyright © 1996, 2003 Oracle. All rights reserved.

Primary Author: Phil Choi

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Send Us Your Comments	vii
Preface	ix
Intended Audience.....	ix
Documentation Accessibility	ix
Conventions	ix
1 Introduction to Oracle Universal Installer	
Key Features	1-1
System Requirements.....	1-2
2 Using Oracle Universal Installer	
Checking Pre-Requisites Before Installation.....	2-1
Installing Oracle Products	2-2
Getting Help While Installing Oracle Products	2-2
About the ORAPARAM.INI File	2-3
Modes of Installation	2-3
Installation Media	2-4
Installing from a Single CD-ROM	2-4
Installing from Multiple CD-ROMs	2-4
TEMP/TMP Directory	2-5
Unmounting a CD	2-5
Installing from the Web	2-5
Special Instructions for UNIX Users	2-6
Failed to Connect to Server Error	2-6
Providing a UNIX Installer Location with Root Privileges	2-6
Providing a UNIX Group Name.....	2-7
Deinstalling Oracle Products	2-7
Removing Oracle Products and Oracle Homes With OUI.....	2-7
Deinstalling Top Level Products that Have Dependees	2-8
Silent Deinstallation.....	2-8
Immediately Displaying the Inventory Dialog Box.....	2-9
Hiding the Inventory Dialog Box	2-9
Hiding the Deinstallation Confirmation and Progress Dialog Boxes	2-9
Running OUI After Installation	2-9

Starting OUI.....	2-9
Command Line Arguments.....	2-10
Using OUI Exit Codes	2-11
Cloning Considerations.....	2-11
About OUI Log Files.....	2-12

3 Managing Oracle Homes

Introduction to Oracle Homes	3-1
Creating Oracle Homes	3-1
Removing Oracle Homes	3-2
Determining the Default Oracle Home	3-2
Multiple Oracle Homes.....	3-3
Target Home	3-3
About the OUI Inventory.....	3-3
Locating the Inventory on Windows Systems	3-3
Locating the Inventory on UNIX	3-4
Home Selector (Available on Win32 Platforms)	3-4
Home Selector Overview	3-4
How Home Selector Works	3-4
Oracle Home Directory Structure for Windows Platforms	3-5
Optimal Flexible Architecture Directory Structure (on UNIX)	3-6
ORACLE_BASE Directory	3-6
ORACLE_HOME Directory	3-6

4 Customizing and Creating Response Files

About Response Files	4-1
What Is a Silent Installation?	4-1
What Is a Response File?	4-1
Why Perform a Silent Installation?	4-2
Modifying a Response File.....	4-2
Creating a Response File With Record Mode	4-3
Using Record Mode	4-3
Response File Format.....	4-3
Variable Values.....	4-4
Variable Lookup Order	4-4
Setting the Recommendation Value	4-4
Comments	4-5
Headers	4-5
Response File Parameters	4-5
Installing Using a Response File.....	4-10
Specifying a Response File.....	4-11
Optional Parameters When Specifying a Response File.....	4-11
Setting Response File Variables From the Command Line.....	4-12
Specifying the Value of a Session Variable	4-12
Specifying the Value of a Component Variable	4-12

5	Cluster Environment Installations	
	Installing Clusterware	5-1
	Pre-Installation Tasks	5-1
	Installing Product Software on a Cluster	5-2
	Cluster Detection.....	5-2
	Availability Checking	5-2
	Cluster Installation.....	5-3
	Cluster Deinstallation	5-3
	Adding Nodes to a Cluster Installation	5-3
	Command Line Options for Cluster Installs (UNIX Only).....	5-4
6	Oracle Internationalization and Translation	
	Installer/Installation Dialogs Language	6-1
	Product Language Selections	6-1
A	Troubleshooting	
	Debugging Mechanisms in OUI	A-1
	OUI Errors.....	A-2
	Other Tips	A-3
B	Sample Files	
	Sample Response File.....	B-1
	Sample ORAPARAM.INI File	B-7
	Index	

Send Us Your Comments

Oracle Universal Installer Concepts Guide, 10g Release 1 (10.1)

Part No. B12140-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: nedc-doc_us@oracle.com
- FAX: 603-897-3317 Attn: Oracle Universal Installer
- Postal service:
Oracle Corporation
Oracle System Management Products Documentation
1 Oracle Drive
Nashua, NH 03062
U.S.A.

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This manual describes how to use the Oracle Universal Installer to install Oracle and third-party software.

Intended Audience

This manual is intended for users installing Oracle software products using the Oracle Universal Installer and covers only the generic functionality and concepts. Use this manual in conjunction with any product-specific installation guides available with your Oracle software product.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)

- [Conventions in Code Examples](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates an element in the user interface.	Click Help for more information.
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
lowercase monospace (fixed-width font)	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter <code>sqlplus</code> to open SQL*Plus. The password is specified in the <code>orapwd</code> file. Back up the datafiles and control files in the <code>/disk1/oracle/dbs</code> directory. The <code>department_id</code> , <code>department_name</code> , and <code>location_id</code> columns are in the <code>hr.departments</code> table. Set the <code>QUERY_REWRITE_ENABLED</code> initialization parameter to <code>true</code> . Connect as <code>oe</code> user. The <code>JRepUtil</code> class implements these methods.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
<>	Angle brackets in command syntax denote an item for which you can substitute a real value. Do not enter the angle brackets.	<code><host>:<port>:<oracle_sid></code>
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	<code>CONNECT SYSTEM/<i>system_password</i></code> <code>DB_NAME = <i>database_name</i></code>

Introduction to Oracle Universal Installer

Oracle Universal Installer (OUI) is a Java-based graphical user interface (GUI) application that enables you to install Oracle components from a CD, multiple CDs, or the Web. OUI performs component-based installations and enables different levels of integrated bundle, suite, and Web-based installations, as well as complex logic in a single package. The installation engine is easily portable across all Java-enabled platforms, and platform-specific issues can be encapsulated from the overall installation process.

OUI provides the following capabilities for addressing software management and distribution:

- Component and suite installs
- Implicit deinstall
- Support for multiple Oracle homes
- National Language Support (NLS)/globalization support
- Automatic dependency resolution and complex logic handling to determine the software to be installed, reinstalled, upgraded, or deinstalled
- Support for distributed installations
- Unattended "silent" installations using response files and/or command line arguments
- The ability to maintain the inventory of installed products

Note: Although OUI can list all products previously installed with another installer (oraInst or ORCA-based), it cannot deinstall them.

Key Features

Oracle Universal Installer 10g Release 1 (10.1) offers the following features:

- **An XML-based centralized inventory.**

The XML format allows third-party Java applications to query the inventory for information about installed software. For backward compatibility, the central inventory continues to maintain the binary versions as well.
- **Cloning of existing Oracle homes.**

Allows you to copy an existing Oracle home to another location and "fix it up" by updating the installation configuration to be specific to the new environment.

Cloning makes it easy to propagate a standard setup without having to install and configure installation after installation.

- **Better support for cluster environments.**

OUI now replicates its inventory to all nodes that participate in a cluster-based installation. Users can invoke OUI from any node on the cluster that is part of the installation. Users can then upgrade/deinstall/patch existing software from any node.

- **True silent capability.**

When running OUI in silent mode on a character mode console, you no longer need to specify an X-server or set the DISPLAY environment variable on UNIX. No GUI classes are instantiated, making the silent mode truly silent.

- **Ability to record your OUI session to a response file.**

This feature makes it easy to duplicate the results of a successful installation on multiple systems. All the options you selected during the installation are saved in the resulting response file. For information about recording your OUI sessions see ["Creating a Response File With Record Mode"](#) on page 4-3.

- **More accurate disk space calculations.**

OUI now uses a more accurate method of calculating the disk space required by your Oracle products. This feature reduces the risk of running out of disk space during an installation.

- **Automatically launched software.**

Some Oracle products now take advantage of a new feature that allows the software to launch automatically immediately after the installation.

- **Cleaner deinstallations and upgrades.**

Deinstallation completely removes all software, leaving no "bits" behind. Also completely removes files associated with configuration assistants and patchsets. Oracle homes can also be removed from the inventory and registry.

- **Integrated Pre-Requisite Checking.**

Provides a pre-requisite checking tool to diagnose the readiness of an environment for installation. The pre-requisite checks are run as part of the installation process, but can also be run as a separate application.

System Requirements

OUI system requirements are listed below:

- **Java Runtime Environment (JRE).** Automatically installed with OUI on most platforms. Check the Release Notes or installation guide of the products you are installing for the required version.
- **Memory Requirements.** Memory requirements vary depending on the number of components installed. Check the Release Notes for the products you are installing for details. 32 MB minimum recommended on all platforms.
- **Disk Space Requirements.** Oracle recommends at least 32 MB for OUI files on Windows machines; 70 MB on UNIX. (UNIX requires more memory because of the difference in JRE sizes for the platforms.) Up to 1 MB may be needed for the related inventory files.

Using Oracle Universal Installer

This chapter includes the following sections:

- [Checking Pre-Requisites Before Installation](#)
- [Installing Oracle Products](#)
- [Deinstalling Oracle Products](#)
- [Running OUI After Installation](#)
- [About OUI Log Files](#)

Checking Pre-Requisites Before Installation

Before installation, OUI checks the environment to see whether it meets the requirements for successful installation. Early detection of problems with the system setup reduces the chances of users encountering problems during installation; for instance, problems with insufficient disk space, missing patches, inappropriate hardware, etc.

OUI is required to perform all pre-requisite checks defined for the installation before installing any software, whether they are OUI-specific tests, or tests defined for a specific product. Specific pre-requisite checks are defined for each operating system on which OUI runs. All pre-requisite check parameters must be defined in the `oraparam.ini` file (or another `.ini` file that you define); all results are logged in the `installActions<timestamp>.log` file.

Pre-requisite checking can be performed in the following three ways:

- **Automatically:** Checks are performed automatically when the user runs the OUI executable during an installation. Simply run OUI and all pre-defined pre-requisite checks will be performed.
- **Silent Mode:** Checks can be run and managed from the command line for a silent installation. For silent installations, OUI performs as many pre-requisite checks as possible, alerts the user to all errors, and provides the location of the `installActions<timestamp>.log` file before exiting.
- **Standalone:** Checks can be run without completing an installation. Refer to [Table 2-1](#) for a description of the flag to use.

Table 2–1 Command Line Parameters for Pre-Requisite Checks

Command	Description
-paramFile <file name>	Allows you to specify the location for the oraparam.ini file to be used. For example, you may create your own parameters file and use that to operate OUI. Note that OUI will use the file you specify for all operations, including the pre-requisite checks.
-executeSysPrereqs	Use this flag on the command line to perform only the pre-requisite check, without continuing the installation. This can be used to verify that the basic pre-requisites have been met before deciding to perform an installation.
-ignoreSysPrereqs	Use this flag on the command line to ignore all pre-requisite checks. This may be useful when you know that the checks themselves are faulty or the environment has already been verified.

Inputs to the checker are listed in the `prerequisite.xml` file. Once the checker is run, results, along with the pre-defined inputs, can be found in the `prerequisite_results.xml` file. These files are located in the `oraInventory/logs` directory. The `prerequisite_results.xml` file can be reused as an input file for subsequent executions of the checker.

Installing Oracle Products

The following sections describe how to start OUI and install an Oracle product. Specifically, this section describes:

- [Getting Help While Installing Oracle Products](#)
- [About the ORAPARAM.INI File](#)
- [Modes of Installation](#)
- [Installation Media](#)
- [Special Instructions for UNIX Users](#)

Getting Help While Installing Oracle Products

At any time while installing your product, click **Help** for information about the screens specific to your installation.

OUI provides two kinds of online help

- Generic online help provided with every copy of OUI.
These topics describe the screens and dialog boxes that every OUI user sees, regardless of the product they are installing.
- Online help specific to a particular installation.
These topics are created by the product developer and describe the screens and dialog boxes specific to the product you are installing. For example, the help topic for the Installation Types page is often a custom help topic created by the installation developer that describes the specific installation types for the product you are installing.

After you display an online help topic, choose **Navigator** from the **Tools** menu to display the navigator pane. From the navigator pane, you can browse the table of

contents, select other topics, or search for a particular word or phrase in the online help.

Note: Only generic help topics are available in the navigator, or table of contents. Custom help topics can only be accessed by clicking the Help button on the dialogs or windows they describe.

About the ORAPARAM.INI File

The `oraparam.ini` file is the initialization file for OUI. This file includes information that defines the behavior of certain OUI features. Each product installation possesses a unique `oraparam.ini` file.

In general, you should not have to edit the contents of this file, but in certain situations, understanding the contents of this file can help you troubleshoot problems and understand certain aspects of the OUI product.

For example, for most installations, OUI provides a default value on the File Locations page that points to the location of the product's installation kit or stage. This default value is stored in the `oraparam.ini` file. The `oraparam.ini` file also identifies the location of the Java Runtime Environment (JRE) required for the installation.

In the staging area, it is located in the same directory as the executable file. For example:

For UNIX systems:

```
cd/Disk1/install/solaris/runInstaller
cd/Disk1/install/solaris/OraParam.ini
```

For Windows systems:

```
cd/Disk1/install/win32/setup.exe
cd/Disk1/install/win32/OraParam.ini
```

In the staging area, the default `OUI_LOCATION` is relative to the location of the `oraparam.ini` file, as follows:

```
../../stage/
```

Once installed, the `oraparam.ini` file is located in the `/oui` directory.

Modes of Installation

You can use Oracle Universal Installer to install Oracle products in any of the three following modes:

- **Interactive:** Use OUI's interactive mode to use the graphical user interface to walk through the installation, providing information in the installation dialogs when prompted. This method is most useful when installing a small number of products in different setups on a small number of machines.
- **Suppressed:** Use OUI's suppressed mode to supply the necessary information by using a combination of a response file or command line entries with certain interactive dialogs. You can choose which dialogs to suppress by supplying the information at the command line when you invoke OUI. This method is most useful when an installation has a common set of parameters that can be captured in a response file, in addition to custom information that must be input by hand.

- **Silent:** Use OUI's silent installation mode to bypass the graphical user interface and supply the necessary information in a response file. This method is most useful when installing the same product multiple times on multiple machines. By using a response file, you can automate the installation of a product for which you know the installation parameters.

See Also: Refer to [Chapter 4, "Customizing and Creating Response Files"](#) for detailed information on using response files and installing in silent mode.

Installation Media

For each of these three installation modes, you can install from three different media:

- [Installing from a Single CD-ROM](#)
- [Installing from Multiple CD-ROMs](#)
- [Installing from the Web](#)

Refer to the following sections to learn more about these different installation approaches.

Installing from a Single CD-ROM

While installing Oracle products contained on a single CD-ROM, start OUI by running the executable file, `setup.exe` or `runInstaller.sh`, located in:

```
../install/<platform>
```

Where `<platform>` represents Win32, Win64, Solaris, Linux, etc.

For UNIX systems, run the script by typing `./runInstaller` at the command line.

Note: OUI for win64 works the same as OUI for win32. However, the startup directory on the CD is "win64" instead of "win32." Launching OUI from the win32 directory launches OUI in 32-bit mode, used for installing 32-bit software. Use win64 for installing 64-bit software.

When you install both 32-bit and 64-bit OUI on a 64-bit machine, two different inventories will be created and maintained. However, you cannot install 64-bit software in a 32-bit home, and vice versa.

Installing from Multiple CD-ROMs

If you are creating a multiple-CD install on UNIX, you may need to launch `runInstaller` in the background using the following command:

```
./runInstaller &
```

By launching `runInstaller` in the background, you can change your current directory after you launch OUI, allowing you to eject the CD. (It may also help to launch `runInstaller` as a foreground process from a different directory.)

You may want to create a shell script that launches OUI in the background and then exits. If you choose to create a shell script, remember to also pass all parameters that passed to the shell script to `runInstaller` in the event that you wish to install silently using a response file.

TEMP/TMP Directory In both UNIX and Windows installations, temporary copies of OUI and JRE are placed in the TEMP or TMP directory in a subdirectory named `/OraInstall<timestamp>` so that these applications can be launched when you change CD-ROMs. Note that temporary files are created for single-CD installations as well.

Note: In both UNIX and Windows, OUI looks for `%TEMP%` then `%TMP%`. If neither is set, OUI will default to `/tmp` on UNIX and `c:\temp` on Windows.

Unmounting a CD On UNIX, If you are having trouble installing a product from multiple CD-ROMs, try using the following procedure to unmount the first CD-ROM and mount the second CD-ROM. If you still have problems, refer to the documentation links at the end of this topic.

In most cases, the following procedure will help with any problems you experience while switching to a second CD-ROM while installing Oracle software. If you inadvertently run the installer while the current working directory is in the CD-ROM, follow these steps to mount the next CD-ROM:

1. Change to the root directory of your system and log in as the root user by using the following commands:

```
$ cd /
$ su root
```

2. Unmount and remove the CD-ROM from the drive with the following command:

```
# umount cdrom_mount_point_directory
```

3. Insert and mount the next CD-ROM into the drive by using the following command:

```
# mount options device_name cdrom_mount_point_directory
```

4. Enter the correct mount point in the Installation dialog box.
5. Click **OK** to continue.

If after attempting this procedure you are still having problems, refer to the section on installing from multiple CD-ROMs in the *Oracle Database Installation Guide*, which is available from the Oracle Technology Network:

<http://otn.oracle.com/documentation>

Installing from the Web

With Oracle Universal Installer, you can install products from Web. You can publish your staging area from a Web server and then in the Universal Installer's **Source** location, specify the HTTP location for the `products.xml` file.

For example, you can enter:

```
http://www.oracle.com/product/ouiinstall/stage/products.xml
```

The Oracle Universal Installer recognizes a Web staging area just like a local, network, or CD-ROM stage.

System administrators of large customers who may want to deploy Oracle software to more than one target can use a combination of the Web installation and response file features:

1. Copy the staging area to a shared file system and make it accessible on the Intranet or a Web server.
2. Include predetermined response files on the same location. (Different groups of users might rely on different response files.)
3. Clients run Oracle Universal Installer locally and use the local response file that is emailed or downloaded so they can perform a silent install.

The Web installation capability relies on some guidelines that must be followed at installation development time. Check your installation guide for your product to see if the installation of your product is certified for Web installation.

To test if your stage is Web-enabled, you may try the following:

1. Copy the stage to your Web server.
2. Start the Oracle Universal Installer locally and point to the location of the `products.xml` file. For example:

```
http://smpweb.us.oracle.com/product/ouiinstall/stage/products.xml
```

Special Instructions for UNIX Users

The following sections describe special instructions that apply when you are installing certain products on a UNIX system.

Failed to Connect to Server Error

If you get an Xlib error or a "Failed to connect to Server" error when you are running OUI on the Solaris operating system, you need to define the following environment variables on the host computer where you are running OUI:

```
%setenv DISPLAY <machine name>:0.0
```

Replace `<machine name>` with the name of the computer that will display OUI.

On the computer that will display OUI, enter the following command. This command allows other computers to display information on the computer's monitor:

```
%xhost +
```

Re-run the `runInstaller` script after you have set the `DISPLAY` environment variable.

Note: You can run OUI without specifying the `DISPLAY` by running in silent mode using a response file.

Providing a UNIX Installer Location with Root Privileges

Various installation operations on the UNIX platform must be performed with root privileges. For example, you must have root privileges to be able to create the OUI inventory.

If you are installing OUI for the first time, you will be prompted to run a shell script from another terminal window before proceeding with the installation. OUI will prompt the user to run `root.sh` once installation completes *only* if the script is required to be run as `root` before configuration assistants are run. Otherwise, users are prompted to run `root.sh` as `root` afterwards.

Note: When running OUI in silent mode, if `root.sh` is required prior to configuration assistants, OUI will skip configuration assistants during the install. The user must run `root.sh` as `root` and then run the skipped configuration assistants after the silent installation is complete.

To successfully run the required shell script:

1. Leave the OUI window open and open another terminal window.
2. In the new terminal window, use the substitute user command to log in with root privileges:

```
su -root
```

3. Change directory to the Oracle home into which you are currently installing your Oracle software product.
4. Run the shell script `./root.sh`.
5. When the script is finished and you are returned to the command prompt, exit from the new terminal window and return to OUI to continue the installation.

Note: Do not exit the installation in order to run the shell script. Exiting the install removes this script.

You are prompted to run the script only the first time you install.

Providing a UNIX Group Name

If you are installing a product on a UNIX system, the Installer will also prompt you to provide the name of the group that owns the base directory.

You must choose a UNIX group name which will have permissions to update, install, and deinstall Oracle software. Members of this group must have write permissions to the base directory chosen.

Only users who belong to this group are able to install or deinstall software on this machine.

Deinstalling Oracle Products

The following sections describe how to deinstall products installed using OUI. Specifically, this section describes:

- [Removing Oracle Products and Oracle Homes With OUI](#)
- [Deinstalling Top Level Products that Have Dependees](#)
- [Silent Deinstallation](#)

Removing Oracle Products and Oracle Homes With OUI

A deinstallation can be performed before selecting products to install or after a successful installation.

To remove an Oracle product or Oracle home using interactive mode, perform the following steps:

1. Start OUI from a CD-ROM or:
 - For Windows platforms, launch OUI from the Start menu by selecting **Start, Installation Products, Oracle Universal Installer**.
 - For UNIX platforms, at the command line, run the script called `runInstaller` from the directory where it is stored, which is by default at the same level as the first Oracle home created on that machine.
2. Click **Deinstall Products** on the "Welcome" screen.

The Inventory panel appears.
3. Select the product(s) you want to remove from the **Contents** tab of the Inventory panel and click **Remove**. Oracle homes may also be removed in the same manner. Once an Oracle home has been removed, you can reuse its name and location to install other products.
4. The Remove Confirmation Dialog appears, asking if you want to remove the products and their dependee components. Click **Yes**.

OUI warns you of any product dependencies that might cause problems if particular products are removed, and prompts you to confirm the de-installation.

Pay special attention to the full list of products being deinstalled before proceeding. OUI computes this list based on the dependencies of each component.

Note: You can also remove products by using the **Installed Products** button on OUI as long as this action is performed before making your selection of products to install.

Deinstalling Top Level Products that Have Dependees

A top level component is the most important component of an installation. It is the installable product you see at the first install screen. You can only install one top level component per install session.

When you select a specific component for deinstallation, OUI analyzes the dependency information to determine if there are other components that should be deinstalled along with it. In general, if a component is selected for deinstall, the following components will be deinstalled with it:

- All components that have a required dependency on the selected component.
- Dependees of the selected component that have no other dependents. A dependee is a component on which the top level component (dependent) has a dependency.

Silent Deinstallation

Not only can you perform command line installations, as described in ["Installing Using a Response File"](#) on page 4-10, you can also perform command line deinstallations. A command line deinstallation allows you to remove Oracle products or Oracle homes from your system without using the OUI graphical user interface.

You can choose to display no dialog boxes or prompts to the user, or you can selectively avoid displaying certain dialog boxes that are normally used during a deinstallation.

Immediately Displaying the Inventory Dialog Box

Use the following commands to immediately display the Inventory dialog box, which allows the user to select items for deinstallation without navigating the OUI startup screen:

```
setup.exe -deinstall (on Windows)
./runInstaller -deinstall (on UNIX)
```

Hiding the Inventory Dialog Box

If you would like to hide the inventory dialog box from the user during a deinstallation, you can specify the products to be deinstalled in the DEINSTALL_LIST parameter of the response file; specify Oracle homes to be removed with the REMOVE_HOMES variable. For more information about response files, see ["Installing Using a Response File"](#) on page 4-10.

Refer to [Chapter 4, "Customizing and Creating Response Files"](#) for information about the DEINSTALL_LIST parameter.

As with other response file parameters, you can also specify the DEINSTALL_LIST parameter on the OUI command line. For example, on a UNIX machine, enter:

```
./runInstaller -deinstall DEINSTALL_LIST={"component1", "1.0.1.2"}
```

To remove Oracle homes from the inventory, use the REMOVE_HOMES variable.

For more information about specifying response file parameters, see ["Setting Response File Variables From the Command Line"](#) on page 4-12.

Hiding the Deinstallation Confirmation and Progress Dialog Boxes

Use the following commands to hide the deinstallation confirmation and progress dialog boxes during a command line deinstallation:

On a Windows system:

```
setup.exe -deinstall session:SHOW_DEINSTALL_PROGRESS=false
session:SHOW_DEINSTALL_CONFIRMATION=false
```

On a UNIX system:

```
./runInstaller -deinstall session:SHOW_DEINSTALL_PROGRESS=false
session:SHOW_DEINSTALL_CONFIRMATION=false
```

Running OUI After Installation

The following sections describe the different ways that OUI can be used after installation. Specifically, this section describes:

- [Starting OUI](#)
- [Command Line Arguments](#)
- [Using OUI Exit Codes](#)
- [Cloning Considerations](#)

Starting OUI

OUI is installed on your system during the installation of your Oracle products:

```
\Program Files\Oracle\oui (on Windows)
At the same level as <oraInventory> (on UNIX)
```

For all platforms, the executable file (`setup.exe` or `runInstaller.sh`) is located in the following directory:

```
<oui_location>/bin
```

A new version of OUI replaces its older version.

OUI is placed under "Independent Products" in the Inventory panel as a "non-Oracle home" product.

To start OUI:

- On Windows platforms, select **Start, Programs, Oracle Installation Products, Oracle Universal Installer**.
- On UNIX, execute `./runInstaller` from the directory where it is installed.
For example: if the `<oraInventory>` is `/u01/oracle/oraInventory`, then, OUI will be at `/u01/oracle/oui`.

A `runInstaller.sh` script is also available, so that users can launch OUI directly from a different directory.

When OUI is first installed and run, it checks for the JRE path (the location from which it runs), using the location specified in the `oraparam.ini` file's `JRE_LOCATION` parameter. If OUI cannot find the JRE specified, an error is returned.

Command Line Arguments

Following is the output from the `runInstaller -help` command, which gives you the full list of command line options and their descriptions, as well as command line variables usage.

Usage:

```
runInstaller [-options] [(<CommandLineVariable=Value>)*]
```

Where options include:

- `-help` Displays above usage.
- `-silent` For silent mode operations, the inputs can be a response file or a list of command line variable value pairs.
- `-responseFile <Path>` Specifies the response file and path to use.
- `-formCluster` To install the Oracle clusterware in order to form the cluster.
- `-remoteShell <Path>` Used only for cluster installs, specifies the path to the remote shell program on the local cluster node.
- `-remoteCopy <Path>` Used only for cluster installs, specifies the path to the remote copy program on the local cluster cluster.
- `-record -destinationFile <Path>` For record mode operation, information is recorded in the destination file path.
- `-deinstall` For deinstall operations.
- `-debug` For getting the debug information from OUI.
- `-ignoreSysPrereqs` For ignoring the results of the system pre-requisite checks.
- `-executeSysPrereqs` Execute system pre-requisite checks and exit.
- `-paramFile` Specify location of `oraparam.ini` file to be used by OUI.
- `-clone` For making an Oracle Home copy match its current environment.
- `-force` Allowing silent mode installation into a non-empty directory.
- `-noconsole` For suppressing display of messages to console. Console is not allocated.
- `-addNode` For adding node(s) to the installation.
- `-removeHome` For removing homes from the OUI inventory.

Command Line Variables Usage

Command line variables are specified using `<name=value>`; for example:

```
[ session: | compName: | compName:version: ]variableName="valueOfVariable"]
```

Session/Installer variables are specified using:

```
[session:]varName=value
Ex 1: session:ORACLE_HOME_NAME="OraHome"
Ex 2: ORACLE_HOME_NAME="OraHome"
The lookup order is session:varName then just varName).
The session prefix is used to avoid ambiguity.
```

Component variables are specified using:

```
[compInternalName: [Version:]]varName
Ex 1: oracle.compl:1.0.1:varName="VarValue"
Ex 2: oracle.compl:varName="VarValue"
Ex 2: oracle.compl:varName="VarValue"
The lookup order is compInternalName:Version:varName, then compInternalName:varName, then
just varName.
```

Using OUI Exit Codes

If you are starting and stopping OUI programmatically (for example, by invoking OUI using a response file), you may need to consider the exit codes generated by OUI and perform a particular action depending on the code OUI returns.

OUI returns one of the following exit codes:

Code	Description
0	All installations were successful.
1	All installations were successful but some optional configuration tools failed.
-1	At least one installation failed.

Note that:

- This feature will not work if OUI is running in "bootstrap" mode. In this case `setup.exe/runInstaller` will just launch the JRE process and return immediately without waiting for the exit code. OUI will be running in "bootstrap" mode if the following line exists in the file `oraparam.ini` file:

```
BOOTSTRAP=TRUE
```
- If you exit without installing any products (for example if you exit from the "Welcome" screen), the exit code will be -1.

Cloning Considerations

You can copy an existing Oracle home, then configure it for its new environment. This process is called "cloning."

Note: Patching and deinstallation on a cloned Oracle home act the same as a regularly installed Oracle home. You may directly patch a cloned installation.

Invoke OUI in clone mode using the following command:

```
./runInstaller -clone ORACLE_HOME="<target location>" ORACLE_HOME_NAME="<unique name on node>" [-responseFile <full path>]
```

Use `setup.exe` instead of `runInstaller` for Windows machines. The `-responseFile` parameter is optional. Clone-time parameters may be supplied on the command line or via the response file named on the command line.

Clone-time activity is logged in the cloneActions<timestamp>.log file at install time

Note: Because most cloning is done in silent mode, when cloning an Oracle home onto a "clean" machine (one that has no oraInst.loc file), OUI creates a central inventory in the location specified by the INVENTORY_LOCATION variable. If this variable is not specified, OUI creates the central inventory in the <cloned_home>/oraInventory directory.

After cloning is finished, you must run oraInstRoot . sh as root to move oraInventory to the final, desired location.

About OUI Log Files

When you install or deinstall products using OUI, important information about each install is saved not only in the inventory, but also in a series of log files, located in the following directory:

```
<central_inventory>/logs
```

These log files can be used to troubleshoot installation problems. These files are also crucial for deinstalling and configuring the various software components you install on your Windows or UNIX computer. OUI displays the name and location of the current session's log file on the Install page.

Note that the logs used to deinstall products are different from the installActions<timestamp>.log generated during the install process. The installActions<timestamp>.log is easier to read and can be used to view the operations performed at install time.

For more information about the log files generated by OUI, refer to the online help. For more information about using the online help, see "[Getting Help While Installing Oracle Products](#)" on page 2-2.

Managing Oracle Homes

This chapter contains the following sections:

- [Introduction to Oracle Homes](#)
- [Creating Oracle Homes](#)
- [Removing Oracle Homes](#)
- [Determining the Default Oracle Home](#)
- [Multiple Oracle Homes](#)
- [About the OUI Inventory](#)
- [Home Selector \(Available on Win32 Platforms\)](#)

Introduction to Oracle Homes

The Oracle Universal Installer (OUI) supports the installation of several active Oracle homes on the same machine. An Oracle home is the system context in which Oracle products run. This context consists of the directory location where the products are installed, the corresponding system path setup, and where applicable, the program groups associated with the products installed in that home, and the services running from that home.

Creating Oracle Homes

To create an Oracle home, follow the steps below.

1. Run OUI.
2. In the File Locations page, enter the Oracle home settings for the installation session under Destination. See [Table 3–1](#) for a description of the fields in this section of the screen.
3. Continue with your installation. Refer to [Chapter 2, "Using Oracle Universal Installer"](#) for detailed information.

Table 3–1 Oracle Installation Settings in Destination

Settings	Functions
Name	Oracle homes are identified by name, and the Oracle home name identifies the program group associated with a particular Oracle home, and the Oracle services installed on the associated home. The Oracle home name must be 1 to 127 characters long and can only include alphanumeric characters and underscores.

Table 3–1 (Cont.) Oracle Installation Settings in Destination

Settings	Functions
Path	<p>Enter an Oracle home and its full path or select an Oracle home from a drop-down list of existing Oracle homes. The Oracle home location is the directory where products are installed.</p> <p>Datafiles may or may not be installed within an Oracle home. You may use the Browse button to choose a directory to install your product.</p> <p>For Windows platforms: It must be a valid path that is not in the Windows directory. Different homes cannot share the same location.</p>

Note: Oracle recommends that you designate an Oracle home location that is an empty or non-existing directory. If you select a directory for the Oracle home location that is not empty or already exists, you will be warned and asked if you wish to proceed.

For silent installations, if a non-empty, existing directory is specified, an error is logged to the console and the `installActions<timestamp>.log` and OUI aborts. To override this condition, use the `-force` flag on the command line. The effect of using the `-force` flag will be the same as selecting **Yes** while installing in GUI mode. You will receive a warning message, but the installation will continue.

Removing Oracle Homes

To remove an existing Oracle home, invoke OUI and click **Deinstall Products**. At the Inventory dialog, select the homes you wish to delete and click **Remove**. You may also use the `REMOVE_HOMES` variable at the command line or in a response file.

Determining the Default Oracle Home

By default, when you start OUI, the software searches your system to determine the default Oracle home where Oracle software should be installed.

In all cases, the `ORACLE_HOME` **name** is taken first from the response file, if specified. If not, then the following convention is used for the name:

```
Ora<short_marketing_name>_<home_identifier>
```

Where `<short_marketing_name>` is the short product marketing name, for example, "Db10g", and `<home_identifier>` is a counter derived from the central inventory. For example, the `ORACLE_HOME` name could be `OraDb10g_1`.

The `ORACLE_HOME` **path** is taken first from the response file, if specified. If not, then the `ORACLE_HOME` environment variable is used. If neither is specified, the following conventions are used for the path:

- If `ORACLE_BASE` *has not* been specified in the environment:

```
$HOME/product/<version>/short_name_<counter>
```

Where `<short_name>` is the short product name, for example, "Db", and `<counter>` is picked up based on the existence of the files. For example, the `ORACLE_HOME` path could be `$HOME/product/10.1.0/Db_1`

- If `ORACLE_BASE` has been specified in the environment:

```
$ORACLE_BASE/product/<version>/short_name_<counter>
```

For example: `$ORACLE_BASE/product/10.1.0/Db_1`.

The instance-related directory location is accepted first from the response file, if specified. If not, the `oradata`, `flash_recovery_area`, `admin`, and `doc` directories are created under `ORACLE_BASE`. If `ORACLE_BASE` has not been specified, the default is the `$ORACLE_HOME/oradata` directory. If the parent directory of the Oracle home is writable, then these directories would be created in the parent directory of the Oracle home.

Multiple Oracle Homes

OUI supports the installation of several active Oracle homes on the same machine as long as the products support this at run-time. You can have multiple versions of the same product or different products running from different Oracle homes concurrently. Products installed in one home will not conflict or interact with products installed in another home. You can update software in any home at any time, assuming all Oracle applications, services, and processes installed on the target home are shut down. Processes from other homes may still be running.

Target Home

The Oracle home currently accessed by OUI for installation or deinstallation is the target home. In order to upgrade or remove products from the target homes, those products must be shut down or stopped.

About the OUI Inventory

The OUI inventory stores information about all Oracle software products installed in all Oracle homes on a machine, provided the product was installed using OUI or `oraInst`.

Inventory information is stored in Extensible Markup Language (XML) format. The XML format allows for easier diagnosis of problems and faster loading of data. Any secure information is not stored directly in the inventory. As a result, during deinstallation of some products, you or your customer may be prompted for required secure information, such as a password.

By default, the OUI inventory is located in a series of directories within the following directories:

```
/Program Files/Oracle/Inventory (on Windows)
<user_home>/oraInventory (on UNIX)
```

Locating the Inventory on Windows Systems

On the Windows platform, OUI first queries the registry to see if the following Oracle home setting is present:

```
HKEY_LOCAL_MACHINE
Software
Oracle
inst_loc
```

If this value is found, it will be used as the inventory location. If the value is not found, and the \$ORACLE_BASE environment variable is not specified, the following sequence occurs:

1. OUI requests the location of the Program Files directory. (The Program Files folder is a system-defined directory usually on the C drive.)
2. The OUI directory structure is created.
3. The registry is updated with this inventory location.

Locating the Inventory on UNIX

At startup, OUI first looks for the following file:

```
var/opt/oracle/oraInst.loc
```

If this value is found, it will be used as the inventory location. If the value is not found, and the \$ORACLE_BASE environment variable is not specified, OUI prompts the user to supply a location for the inventory.

Home Selector (Available on Win32 Platforms)

The following sections describe the Home Selector, which is installed as part of OUI on Windows computers.

To view the Home Selector, click the **Environment** tab of the Inventory dialog, which appears when you click the **Installed Products** button on several OUI screens.

Home Selector Overview

The Home Selector is part of the installation software. The Home Selector enables you to easily change your primary Oracle home (the one that appears first in the PATH environment variable). If you need to switch the active home or need to perform batch work which requires a "default home" to be active, the Home Selector can be used to change the Windows NT system settings.

When using the Home Selector to make a specific \$ORACLE_HOME the active one, the software installation in question is moved to the front of the PATH variable, making it the first directory to be scanned for executables and library files.

Use the GUI in the **Environment** tab of the Inventory dialog to establish the order of Oracle homes in your PATH variable.

How Home Selector Works

When you perform an install on a system, OUI runs the `selectHome.bat` file to register the Oracle home you selected. The first \$ORACLE_HOME will be named the "DEFAULT_HOME" and will register itself in the Windows NT registry under the key:

```
HKEY_LOCAL_MACHINE
  Software
    Oracle
```

This is the default Windows NT registry hive which contains all the "generic" Oracle settings. Also the PATH variable is adjusted and the BIN directory of the \$ORACLE_HOME is added to the environment variable.

Starting with Oracle 8.0.4, the first version to be multiple \$ORACLE_HOME aware, some additional parameters will also be written to the key:

```
HKEY_LOCAL_MACHINE
  Software
    Oracle
      Home0
```

With OUI, you can change the "DEFAULT_HOME" name to something else. Also, it registers all Oracle settings in the "Home0" subkey of the ORACLE key.

When an additional \$ORACLE_HOME is added to the system, the PATH variable is adjusted again to add the new BIN directory of the newly installed Oracle software. The registry variables are written to a key named "HOMEx," where X is the next available number in the Oracle key.

When several \$ORACLE_HOMES are installed, the executables referenced are all found in the first \$ORACLE_HOME\BIN directory of the PATH variable. When you want to switch the order of the PATH variable, you can use the Home Selector to switch the order of the \$ORACLE_HOMES installed on the system.

Together with the installation and registration of the software in the registry, a second series of registry variables are written in the following key:

```
HKEY_LOCAL_MACHINE
  Software
    Oracle
      ALL_HOMES
```

This is the starting point for the list of all \$ORACLE_HOME installations, done on this system.

The following variables are written in this key:

- HOME_COUNTER: Number of Oracle home directories already present on the system. This counter is a number, exactly 1 higher as the last ID subkey present.
- LAST_HOME: ID number of the currently active \$ORACLE_HOME
- IDx: List of registry key's containing the name, number and installation directory of this ORACLE_HOME. The first installation is written in ID0, the next one in ID1, etc.

If the Home Selector is started in interactive mode, the registry hive "ALL_HOMES" is read, including all subkeys, and a dialog box will be displayed with all available installations. As soon as the you select one, the PATH variable will be adjusted at the system level.

The name of this \$ORACLE_HOME is one of the names found in the ID subkeys.

Oracle Home Directory Structure for Windows Platforms

A typical Oracle home on Windows platforms contains the files and directories shown in [Table 3-2](#).

Table 3-2 Oracle Home Directory for Windows Platforms

Oracle Home Directory	Contents
\BIN	Product executables and DLLs
\LIB	DLL files
\JLIB	All JAR files
\DBS	Common message files

Table 3–2 (Cont.) Oracle Home Directory for Windows Platforms

Oracle Home Directory	Contents
\PROD1	PROD1 product files
\PROD2	PROD2 product files

Optimal Flexible Architecture Directory Structure (on UNIX)

The Optimal Flexible Architecture (OFA) standard is a set of configuration guidelines for fast, reliable Oracle databases that require little maintenance.

OFA is designed to:

- Organize large amounts of complicated software and data on disk to avoid device bottlenecks and poor performance
- Facilitate routine administrative tasks such as software and data backup functions, which are often vulnerable to data corruption
- Alleviate switching among multiple Oracle databases
- Adequately manage and administer database growth
- Help eliminate fragmentation of free space in the data dictionary, isolate other fragmentation, and minimize resource contention

The OFA directory structure is described below.

ORACLE_BASE Directory

ORACLE_BASE specifies the BASE of the Oracle directory structure for OFA-compliant databases. A typical ORACLE_BASE directory structure is described in [Table 3–3](#). When installing an OFA-compliant database using OUI, ORACLE_BASE is set to `/pm/app/oracle` by default.

Table 3–3 Sample ORACLE_BASE Directory Structure and Content

Directory	Content
admin	Administrative files
doc	online documentation
local	Subtree for local Oracle software
product	Oracle software

ORACLE_HOME Directory

The following is an example of an Oracle Server Install.

If you install an OFA-compliant Oracle Server, the ORACLE_HOME directory is `/mount_point/app/oracle/product/release_number`. ORACLE_HOME directory structure and content are described in [Table 3–4](#). Under UNIX, the ORACLE_HOME directory might contain the following subdirectories, as well as a subdirectory for each Oracle product selected.

Note: The examples in this table are only samples. The directories that appear in your own Oracle home will depend on whether the corresponding products are installed and the version of the Oracle database you are running.

Table 3–4 Sample Oracle Home Directory Structure and Content

Directory	Content
assistants	configuration Assistants
bin	binaries for all products
ctx	interMedia Text cartridge
db	<i>initsid.ora, lksid</i>
install	install related files
lib	Oracle product libraries
jlib	Java classes
md	Spatial cartridge
mlx	Xerox Stemmer (for interMedia Text cartridge)
network	Net8
nlsrtl	NLS run-time loadable data
ocommon	common files for all products
odg	data gatherer
opsm	Parallel Server Manager Components
oracore	core libraries
ord	data cartridges
otrace	Oracle TRACE
plsql	PL/SQL
precomp	precompilers
rdbms	server files and libraries required for the database
slax	SLAX parser
sqlplus	SQL*Plus

Customizing and Creating Response Files

This chapter introduces you to Oracle Universal Installer's use of response files for silent and suppressed installation. This chapter also describes how to modify or create a response file so you can customize and standardize the installation of Oracle products in your organization.

This chapter is organized into the following sections:

- [About Response Files](#)
- [Modifying a Response File](#)
- [Creating a Response File With Record Mode](#)
- [Response File Format](#)

About Response Files

This section answers the following questions:

- [What Is a Silent Installation?](#)
- [What Is a Response File?](#)
- [Why Perform a Silent Installation?](#)

What Is a Silent Installation?

A silent installation runs in the background and does not require input from the user. The interactive dialogs normally seen by the user are not displayed.

Instead of prompting the user to select a series of installation options, OUI installs the software using a pre-defined set of options stored in a response file or passed on the command line.

What Is a Response File?

A **response file** is a specifications file containing information normally provided by the user through the OUI user interface during an interactive installation session. Each answer is stored as a value for a variable identified in the response file.

For example, values for Oracle home or install type can be set automatically within the response file.

The response file template for the installation of your product can be found on your stage (CD-ROM) under the `<root of CD>/response` directory. For example:

```
<Products.xml_Location>/Response/<product>.<installtype>.rsp
```

Note: Check the installation guide for the product that you are installing to get the correct list of required files.

Why Perform a Silent Installation?

Silent installations can be useful if you have to install an Oracle product multiple times on multiple computers. If the options you select while installing on each computer are always the same, you save the time of reviewing each installation screen and selecting the various installation options.

Silent installations can also ensure that multiple users in your organization use the same installation options when they install your Oracle products. This makes supporting those users easier because you already know what components and options have been installed on each computer.

Before you perform a silent installation, you should review the settings in the response file template provided with your Oracle product.

Note: If you attempt to perform a silent installation on a UNIX computer where no Oracle products have been installed, OUI uses the default inventory location, then prompts you to run the `oraInstRoot.sh` script with root privileges upon successful installation. The script is saved in the `/oraInventory` directory. This script sets up the central inventory on a clean machine.

Refer to "[About the OUI Inventory](#)" on page 3-3 for more information on the centralized inventory.

The rest of this chapter describes the various parameters and settings you can modify within an OUI response file.

Modifying a Response File

If your product installation includes a response file template, you can find it on your stage (CD-ROM) under the `<root of CD>/response` directory.

If your product installation does not include a response file template, you can create a response file based on the installation options you select. Refer to "[Creating a Response File With Record Mode](#)" on page 4-2 for more information.

To modify the response file:

1. Make a copy of the product's response file and open it in a text editor.
2. Review any information provided in the response file or in the product installation guide.

Many software products use settings in the response file to customize the installation of their particular product. Refer to the response file template for suggestions or guidelines on how to set up the file for your installation.

3. Get familiar with the organization and content of the response file using the information in the section "[Response File Format](#)" on page 4-3.
4. Modify the response file to meet the needs of your organization and save the modified version.

5. See the section "Installing Using a Response File" on page 4-10 for information on starting OUI using your modified response file.

Creating a Response File With Record Mode

You can create a new response file, based on the installation options you select, by using OUI's record mode.

When you use record mode, OUI records the installation session into a response file. You specify the name of the response file on the command line. The recorded response file is generated immediately after the Summary page, so you don't need to actually install your Oracle product to create the response file.

In other words, you can start the installation in Record mode and proceed through the installation options until you get to the Summary page. On the Summary Page, click **Exit** to stop the installation from proceeding with the installation. However, all the options you selected will be saved in the resulting response file.

You can use the newly created response file to run identical installation sessions on other computers in your organization.

Record mode can be also used during a silent installation. In those cases, the variable values specified in the original source response file will be recorded into the new response file.

The following sections describe how to use record mode on Windows and UNIX systems.

Using Record Mode

To record a new response file:

1. At the command prompt, use the `cd` command to change to the directory that contains the OUI executable file (`setup.exe` or `runInstaller.sh`) for your installation.
2. Enter the following command:

```
setup -record -destinationFile <response_file_name> (on Windows)
./runInstaller -record -destinationFile <response_file_name> (on UNIX)
```

Replace the `<response_file_name>` with the complete path for the new response file. For example:

```
setup -record -destinationFile C:\response_files\install_oracle910 (on Windows)
./runInstaller -record -destinationFile /private/temp/install_oracle91.rsp (on UNIX)
```

3. Use the OUI user interface to select your installation options. These will be recorded.

When OUI displays the Summary page, you can either continue with the installation or exit.

OUI saves your new response file using the path and file name you specified on the command line.

Response File Format

The following sections describe the organization and content of an OUI response file.

- [Variable Values](#)
- [Comments](#)
- [Headers](#)
- [Response File Parameters](#)

Variable Values

This section contains information on how variables are populated with values.

Variable Lookup Order

All variable values within a response file are in the name-value format. If two components have a variable with the same name, then the expression should be written as follows to preserve each variable's uniqueness:

```
<component>:<variable>=<value>
```

If there are two components with the same internal name, but different versions, the ambiguity is resolved by specifying the expression as follows:

```
<component>:<version>:<variable>=<value>
```

OUI looks for command line variables in the following order:

```
<component>:<version>:<variable>=<value>
<component>:<variable>=<value>
<variable>=<value>
```

If command line variables are not found, then OUI looks for variables in the response file in the same order:

```
<component>:<version>:<variable>=<value>
<component>:<variable>=<value>
<variable>=<value>
```

Note: OUI treats incorrect context, format, or type values within a response file as if no value were specified.

Setting the Recommendation Value

Values for variables are specified as:

```
<variable> = <recommendation> : <value>
```

The values that are given as `<value_required>` must be specified for a silent installation to be successful.

To specify a default value, specify the value as:

```
<variable> = <default> : <value>
```

For values that are given as `<value_unspecified>`, you may optionally specify a value, where `<value>` can be one of the following types listed in [Table 4-1](#).

Table 4-1 Variable Types and Representations

Type	Representation
Number	10

Table 4–1 (Cont.) Variable Types and Representations

Type	Representation
Boolean	TRUE or FALSE (case insensitive)
String	"Value"
String List	{"value1", "value2"}

The <recommendation> parameter can be specified as `Forced` or `Default`.

- If you specify `Forced`, no dialog appears during installation. The value is automatically used. The user does not have the option to change the value.
- If you specify `Default`, the dialog appears during installation with the value as the default. The user has the option to choose another value.
- If nothing is specified for <recommendation>, `Forced` is used as the default.

The format is shown in the following example:

```
InstallType=Default:"Minimum"
```

Comments

Comments begin with a "#" (hash or pound) symbol. They contain information about the type of the variable, state whether the variable appears in dialog, and describe the function of the variable. A variable and a value are associated with a comment.

For example:

```
#TopLevelComponent;StringList;Used in Dialog
TopLevelComponent={"Demo", "1.0"}
```

Headers

The header is a comment that describes the different formats and comment conventions used in a response file.

For example:

```
#Parameter : UNIX_GROUP_NAME
#Type : String
#Description : UNIX group to be set for the inventory directory.
#Valid only on UNIX platforms.
#Example : UNIX_GROUP_NAME = "install"
UNIX_GROUP_NAME=<Value Unspecified>
```

Response File Parameters

The following are parameters for Oracle Universal Installer. All products installed using OUI will have these parameters, in addition to product-specific variables.

INCLUDE

This parameter specifies the list of response files to be included in this response file. If you want to include other response files in your main response file, you must use this parameter. Note that the values in the main response file take priority over the included response files.

```
INCLUDE={"file1.rsp", "file2.rsp", ..... "filen.rsp"}
```

You should specify the absolute path in the INCLUDE statement for each response file to be included. If you wish to specify a relative path, note that the location is relative to the current working directory (the location of the `oraparam.ini` file).

RESPONSEFILE_VERSION

This parameter specifies the version number of the response file.

```
RESPONSEFILE_VERSION = <version_number>
```

FROM_LOCATION

The location of the source of the products to be installed. The default generated value for this parameter is a path relative to the location of the `runInstaller` file. Relative paths are necessary for shared response files used by multiple users over a network. Since people may be mapping to different drives, absolute paths will not work for shared response files.

Notes:

- You must enter a value for `FROM_LOCATION` for a complete silent install. You may want to use the command line to set this parameter. If the location is a relative path, remember that the path should be relative to the location of the `oraparam.ini` file.
 - Make sure that the `/var/opt/oracle/oraInst.loc` exists. The response file for UNIX has a public variable, `setunixinstallgroup`, that only takes effect when it is the first OUI installation and the `oraInst.loc` file is not already present in `/var/opt/oracle`.
-
-

FROM_LOCATION_CD_LABEL

This parameter is used in multiple-CD installations. It includes the label of the compact disk where the file `products.jar` exists. The label can be found in the `disk.label` file in the same directory as `products.jar`.

LOCATION_FOR_DISK2

Complete path to other disks.

```
LOCATION_FOR_DISK2="F:\teststage\cd\Disk2"
```

If there are more than two disks, more variables will be added as `LOCATION_FOR_DISK3`, and so on.

The CD location for a silent installation is located by two mechanisms:

1. OUI looks for the `Location_For_Disk{DiskNumber}` variable in the response file and uses that location.
2. If the variable does not have a value or does not have the required files, it will look for the components under `../.. /Disk{DiskNumber}/stage` (from `products.jar`).

Therefore, you can either specify the location to look for it in the response file or you can copy them into the disk. For example:

```
E:/Disk1/stage  
E:/Disk2/stage
```

RESTART_SYSTEM

Set this boolean variable to `TRUE` if you want to restart the system without the user's confirmation. This is the force value for restarting the system.

NEXT_SESSION

Set this boolean variable to TRUE if the installer needs to go to the File Locations page for another installation or to another response file if you are performing a silent installation.

```
NEXT_SESSION = TRUE
```

NEXT_SESSION_ON_FAIL

Set this boolean variable to TRUE to allow users to invoke another installation session even if the current installation session fails. This variable is used only if the NEXT_SESSION variable is set to TRUE.

```
NEXT_SESSION_ON_FAIL = TRUE
```

NEXT_SESSION_RESPONSE

Use this string variable to enter the complete path of the next session's response file if you want to automatically begin another silent installation when the current installation is complete. If you specify only a file name, the Installer looks for the response file in the <TEMP>/orainstall directory. This variable is only used if NEXT_SESSION is set to TRUE; otherwise, OUI ignores the value of this variable.

```
NEXT_SESSION_RESPONSE="/private/usr2/nextinstall.rsp"
```

ORACLE_HOME

The location where products are to be installed. You must enter a value for ORACLE_HOME for a complete silent install.

ORACLE_HOME_NAME

The name of the current Oracle home name. You must enter a value for ORACLE_HOME_NAME for a complete silent install.

SHOW_COMPONENT_LOCATIONS_PAGE

The location page, which appears in a custom install type, can be suppressed by setting this value to FALSE. If you set the value to FALSE, the user is prevented from specifying alternate directories. If there are products with installed directories which can be changed, you may want to set the value to TRUE.

SHOW_CUSTOM_TREE_PAGE

Set SHOW_CUSTOM_TREE_PAGE to TRUE if the custom tree page in the installer must be shown. In the Custom Tree page, dependencies can be selected or de-selected. This page appears only in a custom install type.

SHOW_END_SESSION_PAGE

The install success/failure page, which appears at the end of an installation, can be suppressed by setting this value to FALSE.

SHOW_EXIT_CONFIRMATION

Set to TRUE if the confirmation when exiting the installer needs to be shown.

```
SHOW_EXIT_CONFIRMATION = TRUE
```

SHOW_INSTALL_PROGRESS_PAGE

The install progress page, which appears during the install phase, can be suppressed by setting the value to FALSE.

SHOW_OPTIONAL_CONFIG_TOOL_PAGE

Set to TRUE if the Optional Configuration Tools page in the installer must be shown. The Optional Configuration Tools page shows the list of optional configuration tools

that are part of this installation and the status of each tool, including detailed information on why the tool has failed.

SHOW_ROOTSH_CONFIRMATION

Set to TRUE if the Confirmation dialog asking to run the `root.sh` script in the installer needs to be shown. This variable is valid only on UNIX platforms.

```
SHOW_ROOTSH_CONFIRMATION = TRUE
```

SHOW_SPLASH_SCREEN

Set to TRUE if the initial splash screen in the installer needs to be shown.

```
SHOW_SPLASH_SCREEN = TRUE
```

SHOW_SUMMARY_PAGE

The summary page can be suppressed by setting this value to FALSE.

SHOW_WELCOME_PAGE

Set to TRUE if the Welcome page in the installer needs to be shown.

```
SHOW_WELCOME_PAGE = FALSE
```

SHOW_RELEASE_NOTES

Set this parameter to TRUE if you want the release notes for this installation to be shown at the end of the installation. A dialog box lists the available release notes. Note that the `SHOW_END_SESSION` parameter must be set to TRUE before you can use this parameter.

TOPLEVEL_COMPONENT

The name of the component (products) and the version as string list. You must enter a value for `TOPLEVEL_COMPONENT`.

Usually the components are represented with a pair of strings: the first one representing the internal name, the second representing the version.

For example, RDBMS 9.2. may be represented as
{ "oracle.rdbms", "9.2.0.4.0" }.

UNIX_GROUP_NAME

The UNIX group name to be set for the inventory in UNIX platforms.

Note: The UNIX group name is used for first-time installations only.

REMOVE_HOMES

Use this parameter to identify the Oracle homes you wish to remove from the inventory during a deinstallation session. For each home, specify the home name, using full path information.

```
REMOVE_HOMES={ "/home/oracle/ora9i", "/home/oracle/ora8i" }
```

DEINSTALL_LIST

Use this parameter to enter a list of components to be deinstalled during a silent deinstall session. For each component, specify the internal component name and version.

```
DEINSTALL_LIST={ "sample1", "1.0.0.0.0" }
```

SHOW_DEINSTALL_CONFIRMATION

Set this parameter to FALSE if you want to hide the deinstall confirmation dialog box during a silent deinstallation.

SHOW_DEINSTALL_PROGRESS

Set this parameter to FALSE if you want to hide the deinstallation progress dialog box during a silent deinstallation.

DEPENDENCY_LIST

The DEPENDENCY_LIST is the list of dependees on which the component depends. These dependee components represents the list of components that appear as "selected" during installation. Following is a list of some of this parameter's characteristics:

- The DEPENDENCY_LIST variable is only generated when dependencies are present and if the dependency is not a required one.
- You cannot list components to appear as de-selected items in a dialog.
- Specifying required dependees is redundant since they will be selected anyway. The dependees selection can have the other two types of dependees: optional and one-or-more.
- The list of components is specified by an internal name and version number.

A typical example is shown below:

```
DEPENDENCY_LIST={"oracle.netclt", "9.2.0.4.0", "oracle.netmgr", "9.2.0.4.0"}
```

CLUSTER_NODES

This parameter lists the nodes on the cluster on which to install during a silent installation. OUI will install on all named nodes.

```
CLUSTER_NODES={"alpha-1", "alpha-2"}
```

OPTIONAL_CONFIG_TOOLS

While all the required configuration tools are launched by the installer, you can control the configuration tools you would want to launch by specifying the tool's internal names in the OPTIONAL_CONFIG_TOOLS section.

Note: The OPTIONAL_CONFIG_TOOLS variable is only generated when at least one optional configuration tool is available.

You can specify both the Auto-launch optional tools and User-launch optional tools in a string list.

```
#Example : OPTIONAL_CONFIG_TOOLS = {"configtool2 ", "configtool3"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>
```

If no value is specified for this variable, then all the tools are run by default.

If there is a value specified, then only those tools are run while the rest of the tools are ignored.

The way to suppress the configuration tool is to mention only the tools that you want to run as part of the OPTIONAL_CONFIG_TOOLS variable added per component. You should use the internal names of the configuration tool. The response file generator generates these internal names also as part of the options given for the variable.

For example, if `oracle.server` has Tool1 and Tool2 and you want to run only Tool1 in the response file, you mention it as follows:

```
oracle.server:10.1.0.2.0:OPTIONAL_CONFIG_TOOLS={"Tool1"}
```

INSTALL_TYPE

You can set the install type variable to determine the install type of the currently selected top level component.

The install type variable is only generated for the top level components and only when there are more than one install type available.

Note: You must enter a value for INSTALL_TYPE.

SELECTED_LANGUAGES

You can set the languages in which the components will be installed. You must use the internal name while specifying the value:

en, : English
fr, : French
de, : German
ja, : Japanese

For example, to specify Japanese:

```
SELECTED_LANGUAGES = {"ja"}
```

For more information on the languages OUI supports, refer to "[Product Language Selections](#)" on page 6-1.

Note: The SELECTED_LANGUAGES variable is only generated when more than one language is available.

ACCEPT_LICENSE_AGREEMENT

Set this Boolean variable to TRUE if you agree with the license agreement. This parameter requires you to set this variable to TRUE to indicate your acceptance of the license agreement normally accepted in the GUI. This variable is used only in silent installations. Installation cannot continue until this variable is set to TRUE.

Installing Using a Response File

Many Oracle software products provide tools and procedures for running OUI from the command line without displaying OUI screens or responding to questions during the installation.

These are called silent installations. Instead of prompting you to select a series of installation options, OUI installs the software using a predefined set of options. These options are stored in a response file (.rsp).

Consider the following information about response files:

- If your product installation includes a response file, you can find it on your stage (CD-ROM) under the <root of CD>/response directory.
- You can modify the response file for your Oracle product to customize an installation for your organization. Refer to [Chapter 4, "Customizing and Creating Response Files"](#) for more information.
- You can create your own response files using record mode. Refer to [Chapter 4, "Customizing and Creating Response Files"](#) for more information.
- If you start OUI from the command line, see "[Using OUI Exit Codes](#)" on page 2-11.

The following sections describe how to specify a response file when you start OUI.

Note: If you attempt to perform a silent installation on a UNIX computer where no Oracle products have been installed, you will get an error message. Before you can perform a silent installation on such a computer, you must first run the script `oraInstRoot.sh`, which is saved in the `/oraInventory` directory. You must run this script with root privileges. This enables OUI to set up the central inventory on a clean machine.

Refer to "[About the OUI Inventory](#)" on page 3-3 for more information on the centralized inventory.

Specifying a Response File

To start OUI and specify the response file, enter the following command at the command line in the directory where the executable file is installed:

```
setup.exe -responseFile <filename> <optional_parameters> (on Windows)
./runInstaller -responseFile <filename> <optional_parameters> (on UNIX)
```

Note: You must specify the complete `responseFile` path. If you do not, OUI assumes the location to be relative to the `oraparam.ini` associated with the OUI launched.

For help on command line usage, enter the following at the command line in the directory where the executable file is stored:

```
setup -help (on Windows)
./runInstaller -help (on UNIX)
```

Optional Parameters When Specifying a Response File

Optional parameters you may use with the `-responseFile` flag are:

```
-nowelcome
-silent
-fromCluster
```

Use the `-nowelcome` flag along with the `-responseFile` flag to suppress the Welcome dialog that appears during installation.

Use the `-silent` flag along with the `-responseFile` flag to run OUI in complete silent mode. Note that the Welcome dialog is suppressed automatically.

Use the `-fromCluster` flag for clusterware installations to specify the cluster.

Note that when the `-silent` flag is specified and there are no values specified to a particular variable in a dialog, then the installer will abort. The success or failure of the installation, when this flag is specified, is generated:

- In a file named `silentInstall<timestamp>.log` for machines without an Oracle inventory. This file will be generated in the `/tmp` directory in UNIX and the directory specified by the variable `TEMP` in Windows platforms.
- In the inventory logs directory for machines that already had an inventory.

Note: Using the `-nowelcome` option with the `-silent` option is unnecessary since the Welcome screen does not appear when you use the `-silent` option.

Setting Response File Variables From the Command Line

With OUI 2.1 and higher, you can specify the value of certain variables when you start OUI from the command line. Specifically, you can specify session and component variables. For specific information about the format and organization of response files, see "[Modifying a Response File](#)" on page 4-2.

When you specify the value of a variable on the command line, that value overrides the value of the variable if it is defined in the response file.

Specifying the Value of a Session Variable

To specify the value of a session variable, use the following command syntax:

```
./runInstaller session:<variable_name>=<value> (on UNIX)
setup.exe session:<variable_name>=<value> (on Windows)
```

For example, to prevent the Universal Welcome page from displaying:

```
./runInstaller session:SHOW_WELCOME_PAGE=false (on UNIX)
setup.exe session:SHOW_WELCOME_PAGE=false (on Windows)
```

Note that the "session:" tag is optional and is used mainly to remove any possible ambiguity.

Specifying the Value of a Component Variable

To specify the value of a component variable, use the following command syntax:

```
./runInstaller <component_name>:<component_version>:<variable_name>=<value> (on UNIX)
setup.exe <component_name>:<component_version>:<variable_name>=<value> (on Windows)
```

For example, to modify the value of a the variable VAR1 in version 1.1 of a component called COMP2:

```
./runInstaller COMP2_1.1_VAR1="test" (on UNIX)
setup.exe COMP2_1.1_VAR1="test" (on Windows)
```

Note that the "<component_name>:" and "<component_version>:" tags are optional and are used mainly to remove any possible ambiguity. If there are two variables with the same name for different components/versions, then use these tags to distinguish between them.

Cluster Environment Installations

A cluster installation uses Oracle Universal Installer to install software on the nodes of a cluster that are network reachable and bound together by Oracle clusterware. You can use OUI to extend the Oracle home of a product installation to include additional nodes on the cluster.

This chapter includes the following sections:

- [Installing Clusterware](#)
- [Pre-Installation Tasks](#)
- [Installing Product Software on a Cluster](#)
- [Adding Nodes to a Cluster Installation](#)
- [Command Line Options for Cluster Installs \(UNIX Only\)](#)

Installing Clusterware

Clusterware contains components to control the operating system and clusterware services required for managing the nodes on the cluster. In order to create the cluster, you must install Oracle Clusterware software (for example, CRS) before you can install the product software (for example, RAC). During the clusterware installation, OUI prompts you to enter the nodes you wish to include in the cluster in the Cluster Configuration Information page.

After the cluster is formed, you can install cluster-ready Oracle products on the cluster; for instance, an Oracle Real Application Clusters Database or Oracle Application Server infrastructure.

See Also: For details, refer to your clusterware documentation and the installation guide for the cluster-ready product you wish to install. These documents may be found at:

<http://otn.oracle.com/documentation>

Pre-Installation Tasks

Before you can install in a cluster environment, you may need to perform the following pre-installation tasks on each node that is part of the cluster:

- Create the osdba group.
- Create the oracle user.

- Set up user equivalence (for example, `rsh` and `rcp` on UNIX, `net use` on Windows). Users must be able to perform remote operations without requiring credentials information.
- Make sure all nodes are network reachable.
- If vendor clusterware is installed, check if the cluster is configured properly by running `lsnodes`. Type `lsnodes -help` at the command line for details.

See Also: Refer to the specific pre-installation requirements for your product. For RAC installs, refer to the *Real Application Clusters Installation and Configuration* guide.

Installing Product Software on a Cluster

Once clusterware has been installed and pre-installation tasks have been completed, you are ready to install the cluster-ready product on the cluster. To do this, you may use OUI in interactive mode, command line mode, or silent mode using a response file.

See Also: Refer to "[Modes of Installation](#)" on page 2-3 for more information on the different modes.

Cluster Detection

When OUI detects that the destination you entered on the File Locations page is part of a cluster, one of two dialogs will appear:

- If the Oracle home name entered is new, the Specify Hardware Installation Mode page appears. Use this page to select between cluster and non-cluster installation:
 - **Cluster installation:** Select the nodes for your installation from the list of nodes on which clusterware has been installed.
 - **Non-cluster installation:** Install on a single-node only, even though that node is part of a cluster.
- If the Oracle home name entered is a pre-existing home, the Selected Nodes page appears. This is an information-only page that displays the nodes associated with the Oracle home. The installation will be performed on all nodes selected. If any of the selected nodes are down, then the installation will not be allowed to proceed.

Note: In OUI, click **Installed Products** to display the Inventory panel, which lists all installed Oracle homes. Clustered Oracle homes, homes installed on multiple cluster nodes, have an attribute called Cluster Nodes, which lists all the nodes associated with that home.

Availability Checking

When the user clicks **Next** on the Specify Hardware Installation Mode page or Selected Nodes page, OUI performs the following availability checks on the nodes:

- **Network reachability:** Tests whether the remote cluster nodes are up and running.
- **Network configuration:** Tests whether the remote cluster nodes are properly network configured.

- **Inventory setup:** Tests whether the inventory is set up on the remote nodes. If not set up, on UNIX, OUI prompts the user to run root scripts on the remote nodes to set up inventory. On Windows, OUI sets up the inventory location in the Windows registry.
- **Inventory permission:** Tests whether the central inventory location is writable by the user.
- **Oracle home permission:** Tests whether the Oracle home is writable by the user.

If all selected nodes are available, the installation process continues to the next step. If a node (or nodes) is not available, the page is re-displayed with a **Status** column indicating the results of the check. The user must fix the problem or choose another set of nodes in order to proceed.

If any of the remote nodes is unreachable, check if those nodes are up and running and if they are properly network configured. If either the inventory or the Oracle home is not writable, check for the appropriate permissions on the remote nodes for these directories.

Cluster Installation

In a typical cluster installation, when clusterware is present, Universal Installer installs the Oracle software onto the node on which OUI is running, then propagates the Oracle home from the local node installation to the other nodes that are part of the installation.

If the cluster is Cluster File System (CFS) or Network File System (NFS) mounted, then OUI does not propagate the Oracle home to other nodes, as files are shared across nodes and the installation will exist on the shared disk.

Note: For cluster installations, you must run `oraInstRoot.sh` on each node of the cluster to set up the inventory.

Cluster Deinstallation

When deinstalling a clustered Oracle home, OUI first removes software from the node from which you are deinstalling, then removes software from the other cluster nodes associated with the Oracle home.

Adding Nodes to a Cluster Installation

Use OUI to extend the Oracle home of a product installation to include additional nodes on the cluster. Extending an Oracle home to include new nodes is a two-step process:

- First, add the new nodes to the clusterware Oracle home.
- Next, add the new nodes to the product Oracle home.

To add new nodes to either the clusterware or product Oracle home, invoke the `addNode` script from the Oracle home of an existing node you wish to extend. Do not invoke OUI from the CD when adding nodes. At the command line, run the `addNode` script as follows:

```
addNode.sh (UNIX)
addNode.bat (Windows)
```

Adding a node copies the necessary software to the new node and configures the new Oracle home. You can invoke the `addNode` script in silent installations using the `-addNode` flag.

If you have already installed the Oracle home in the correct location on the remote nodes and wish to group that home with an existing cluster Oracle home in the inventory without copying software from one node to the others, invoke the `addNode` script with the `-noCopy` option.

```
addNode.sh -noCopy (UNIX)
addNode.bat -noCopy (Windows)
```

Command Line Options for Cluster Installs (UNIX Only)

When using OUI in suppressed or silent mode, specify the path to the remote copy or shell program on the local node to use for cluster installs using the following optional flags:

```
-remoteShell <path to program>
-remoteCopy <path to program>
```

The default for `remoteShell` is `/usr/local/bin/ssh`. The default for `remoteCopy` is `/usr/local/bin/scp`. If these are not present, OUI defaults to `rsh` and `rcp` respectively.

Oracle Internationalization and Translation

The following sections describe how OUI supports multiple languages and internationalization:

- [Installer/Installation Dialogs Language](#)
- [Product Language Selections](#)

Installer/Installation Dialogs Language

OUI runs in the language of the operating system. OUI uses the language that Java detects, the system locale value, and sets that to the default language. The OUI dialogs will be displayed in this language if available. If specific OUI dialogs are not translated in the language of the operating system, those dialogs will be shown in English.

OUI displays the translated GUI only if the variable `NLS_ENABLED` has been set to `TRUE` in the `oraparam.ini` file. If the `NLS_ENABLED` variable is set to `FALSE`, all text is shown in English.

Product Language Selections

If multiple languages are defined for the products installed, OUI allows language selection for all install types via the **Languages** button (shown on the Available Products page when there is more than one top level component).

Note: If there is only one top level component, the **Languages** button is shown on the Install Types page.

If the staging area has only one top level component with only one install type, then the button is not shown and a Language Selection dialog is displayed later in the installation. You can also control the selected languages by setting the `SELECTED_LANGUAGES` variable in a response file or through the command line:

```
setup.exe SELECTED_LANGUAGES={"fr","de"}
```

The language selections are for the top level component being installed. A top level component is the product selected on the Products Selection dialog, one per installation session. If any sub-components (not top level) do not have files defined for the language the user selects, English only will be installed. The English files are always installed for all components, regardless of user selection or operating system language.

At install time, OUI looks at the language list of each file and file group to determine which files need to be copied.

The OUI product is translated into 9 languages and includes translations for all generic OUI dialogs. Note that any custom dialogs and product-specific information that are part of your installations must be translated and staged independently.

Table 6–1 *OUI is translated into these languages*

Language	Abbreviation
French	fr
Brazilian Portuguese	pt_BR
German	de
Italian	it
Japanese	ja
Korean	ko
Simplified Chinese	zh_CN
Spanish	es
Traditional Chinese	zh_TW

Troubleshooting

This appendix is organized into the following troubleshooting sections:

- [Debugging Mechanisms in OUI](#)
- [OUI Errors](#)
- [Other Tips](#)

Debugging Mechanisms in OUI

During the install, OUI writes a text file that contains information on:

- Variable settings
- Action calls
- Queries
- Exception information

The log of installation actions is written to a file named `installActions<date>.log`, located in the `<oraInventory>\logs` directory. The `oraInstall<timestamp>.err` and `oraInstall<timestamp>.out` files are also created and stored in the same logs directory. The `.log`, `.err`, and `.out` file for a particular session will be named with the same timestamp.

The installation log is used for debugging purposes only. All actions, queries, everything that happens during the install, and all modifications to the target machine are logged in the file.

A sample file is shown below:

```
installActions2002-08-16_09-52-16-AM.log
Environment variables:
ORACLE_HOME =
PATH = E:\OEM\bin;E:\ORACLE815\bin;C:\Program
Files\Oracle\jre\1.1.7\bin;C:\ORATST\BIN
CLASSPATH = C:\Program Files\Exceed.nt\hcljrcsv.zip;
Username is: jdoe
```

```
The installer version is 2.2.1.0.0
```

```
*** Welcome Page***
Setting value of FROM_LOCATION to E:\bootstrap\cd\Disk1\stage\products.jar
Setting value of INVENTORY_LOCATION to C:\Program Files\Oracle\Inventory
Setting value of UNIX_GROUP_NAME to
Setting value of FROM_LOCATION to E:\bootstrap\cd\Disk1\stage\products.jar
Setting value of ORACLE_HOME to E:\Universal
```

```
Setting value of ORACLE_HOME_NAME to Universal

*** File Locations Page***
FromLocation = FROM_LOCATION = E:\bootstrap\cd\Disk1\stage\products.jar
ToLocation = ORACLE_HOME = E:\Universal
ToName = ORACLE_HOME_NAME = Universal
Initializing installer access setup
Setting value of ORACLE_HOME to E:\Universal
Setting value of ORACLE_HOME_KEY to Software\ORACLE\HOME3
Setting value of ORACLE_HOME_FOLDER to Oracle - Universal
Setting value of ORACLE_HOME_SERVICE to Universal
Setting value of ToplevelComp to oracle.swd, 2.2.1.0.0, >0.0, [ 912 453 615 50 601
173 467 295 87 610 198 918 913 162 2 30 21 10021 111 90 168 888 ] [OH:0]
Setting value of SELECTED_LANGUAGES to [en]
Doing operation for installer access setup
Initializing inventory setup WCCE
Doing operation in inventory setup WCCE

*** Available Products Page***
TopLevelComp = ToplevelComp = oracle.swd, 2.2.1.0.0, >0.0, [ 912 453 615 50 601
173 467 295 87 610 198 918 913 162 2 30 21 10021 111 90 168 888 ] [OH:0]
LangsSel = SELECTED_LANGUAGES = [en]
Setting value of ToplevelComp to oracle.swd, 2.2.1.0.0, >0.0, [ 912 453 615 50 601
173 467 295 87 610 198 918 913 162 2 30 21 10021 111 90 168 888 ] [OH:0]
Setting value of DepMode to Complete
Setting value of TLDepModes to Complete,

*** Installation Types Page***
DepMode = DepMode = Complete
TLDepModes = TLDepModes = Complete,
Setting value of DepMode to Complete
Setting value of PROD_HOME to E:\Universal\oracle.swd
Setting value of PROD_HOME to E:\Universal\oracle.swd
Setting value of PRE_REQUISITE to true
Setting value of PROD_HOME to E:\Universal\oracle.swd
Setting value of SHOW_COMPONENT_LOCATIONS_PAGE to true

*** Component Locations Page***
ShowCompLocs = SHOW_COMPONENT_LOCATIONS_PAGE = true
Entering component: oracle.swd installation

*** Summary Page***
ShowSummary = SHOW_SUMMARY_PAGE = null
Global Settings
Source : E:\bootstrap\cd\Disk1\stage\products.jar
Destination : E:\Universal
```

Text files are also written in the temp directory:

```
<temp>\OraInstall\oraInstall<timestamp>.err
<temp>\OraInstall\oraInstall<timestamp>.out
```

If a problem occurs during the installation, you should look for Java exceptions in these files.

OUI Errors

Refer to the release notes for information on any OUI limitations for a particular version.

The most common OUI errors are listed below:

- Out of "temp" space

Make sure you have enough space in the default TEMP or TMP directory otherwise the installation will not succeed. The amount of space required depends on the product being installed.

Note: In both UNIX and Windows, OUI looks for %TEMP% then %TMP%. If neither is set, OUI will default to /tmp on UNIX and c:\temp on Windows.

- Incomplete stage and missing files in file groups

Make sure you have a good staging area. You may have incorrectly copied the staging area from the CD to the hard disk.

- Exceptions from action libraries:

- NT Services, if a service is already started
- File permissions

- UNCAUGHT_RUNTIME exception when user tries to install the product on his machine

This is a blanket error catch that catches any run-time errors like NullPointers from libraries. Check if the temp\OraInstall\orainstall.err has any stack trace. The user will have to call support to debug the staging area.

Other Tips

The following tips may help you to troubleshoot:

- Make sure the correct version of the JRE is specified in the PATH.
- If Java cannot run applications from a Sun machine on an NCD X terminal (OUI does not start), remove the following file:

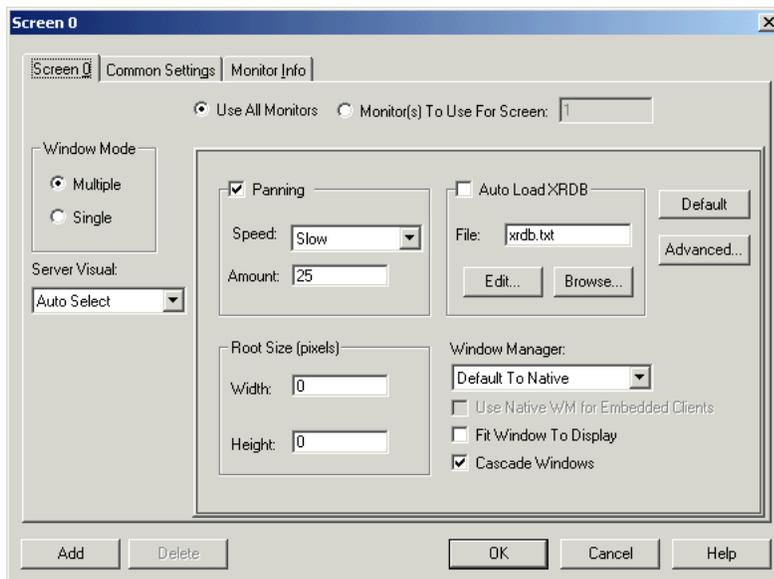
```
$JAVA_HOME/lib/font.properties
```

- If you are deleting an Oracle home manually without using OUI, the products remain registered with OUI. You must then simulate a deinstall so that all OUI references are removed.
- If the install is canceled:
 - Products will not be registered with OUI
 - Some files may have been copied, depending on when you issued the cancellation.

X terminal emulators that were tested with OUI are listed in the following sections.

Exceed

There is a known compatibility issue that Hummingbird has identified to be a problem with Exceed. You can fix it by going into XConfig/Screen Definition/Screen 0 and changing Window Manager from "Default to Native" to "Native." Refer to [Figure A-1](#) to see the proper setup.

Figure A-1 Setting Exceed's Window Manager to "Native"

Another known issue where Exceed fails to recognize fonts is documented at the following site:

<http://developer.java.sun.com/developer/bugParade/bugs/4262023.html>

Reflection X

- Reflection X 6.2+: Select Settings/Window Manager/ Default Local Window Manager/ Reflection Window Manager. You may also have to select "X terminal desktop."
- Reflection X 6.0: Select Connection/New XDMCP Connection/Connection Settings/Direct method/ type host destination and click **Connect**.

Tarantella

There is a known problem when using Tarantella to send DISPLAY to your Windows system, for instance, when using hosted UNIX systems. The bottoms of some OUI screens may appear "cut off," with buttons not visible. The problem occurs only when you use an individual Tarantella terminal window so that that OUI window appears alone without a broader X-windowing environment.

To work around this problem, do one of the following:

- Use keyboard shortcuts to invoke commands:
 - Use **Alt-N** for **Next** to move to the next screen.
 - On the summary page, use **Alt-I** to start the install.
- Use the Window Manager window in Tarantella, which displays the entire screen as if it were a UNIX monitor. Installer dialogs will appear properly in Window Manager.

Sample Files

This appendix provides the following sample files:

- [Sample Response File](#)
- [Sample ORAPARAM.INI File](#)

Sample Response File

Response files are of type `.rsp` and are used by OUI to run silent installations.

```
#####
## Copyright (c) Oracle Corporation 1998,2003. All rights reserved.##
##
## Specify values for the variables listed below to customize   ##
## your installation.                                         ##
##
## Each variable is associated with a comment. The comment    ##
## identifies the variable type.                               ##
##
## Please specify the values in the following format:         ##
##
##      Type          Example                                     ##
##      String        "Sample Value"                           ##
##      Boolean       True or False                             ##
##      Number        1000                                      ##
##      StringList    {"String value 1","String Value 2"}      ##
##
## The values that are given as <Value Required> need to be   ##
## specified for a silent installation to be successful.      ##
##
## This response file is generated by Oracle Software         ##
## Packager.                                                  ##
#####

RESPONSEFILE_VERSION=2.2.1.0.0

#-----
#Name      : UNIX_GROUP_NAME
#Datatype  : String
#Description: Unix group to be set for the inventory directory. Valid only in Unix platforms.
#Example   : UNIX_GROUP_NAME = "install"
#-----
UNIX_GROUP_NAME=<Value Unspecified>
#-----
#Name      : FROM_LOCATION
#Datatype  : String
#Description: Complete path of the products.jar.
#Example   : FROM_LOCATION = "../stage/products.jar"
#-----FROM_
```

```
LOCATION=../../stage/products.jar"
#-----
#Name      : FROM_LOCATION_CD_LABEL
#Datatype  : String
#Description: This variable should only be used in multi-CD installations. It includes the
label of the Compact Disk where the file "products.jar" exists. The label can be found in the
file "disk.label" in the same directory as products.jar.
#Example : FROM_LOCATION_CD_LABEL = "CD Label"
#-----
FROM_LOCATION_CD_LABEL="LABEL1"
#-----
#Name      : NEXT_SESSION_RESPONSE
#Datatype  : String
#Description: Optionally specifies the full path of next session's response file. If only a
file name is specified, the response file is retrieved from <TEMP>/oraInstall directory. This
variable is only active if NEXT_SESSION is set to true.
#Example : NEXT_SESSION_RESPONSE = "nextinstall.rsp"
#-----
NEXT_SESSION_RESPONSE=<Value Unspecified>
#-----
#Name      : ORACLE_HOME
#Datatype  : String
#Description: Complete Location of the Oracle Home.
#Example : ORACLE_HOME = "C:\OHOME1"
#-----
ORACLE_HOME=<Value Required>
#-----
#Name      : ORACLE_HOME_NAME
#Datatype  : String
#Description: Oracle Home Name. Used in creating folders, services.
#Example : ORACLE_HOME_NAME = "OHOME1"
#-----
ORACLE_HOME_NAME="OHOME1"
#-----
#Name      : TOPLEVEL_COMPONENT
#Datatype  : StringList
#Description: The Toplevel component that has to be installed in the current session.
#The following choices are available. The value should contain only one of these choices.
#The choices are of the form Internal Name, Version : External name. Please use the internal
name and version while specifying the value.
#   oracle.swd, 2.2.1.2.0 : Oracle Installation Products 2.2.1.2.0
#Example : TOPLEVEL_COMPONENT = {"oracle.swd", "2.2.1.2.0"}
#-----
TOPLEVEL_COMPONENT={"oracle.swd", "2.2.1.2.0"}
#-----
#Name      : DEINSTALL_LIST
#Datatype  : StringList
#Description: List of components to be deinstalled during a deinstall session.
#The following choices are available. The value should contain only one of these choices.
#The choices are of the form Internal Name, Version : External name. Please use the internal
name and version while specifying the value.
#   oracle.swd, 2.2.1.2.0 : Oracle Installation Products 2.2.1.2.0
#Example : DEINSTALL_LIST = {"oracle.swd", "2.2.1.2.0"}
#-----
DEINSTALL_LIST={"oracle.swd", "2.2.1.2.0"}
#-----
#Name      : SHOW_SPLASH_SCREEN
#Datatype  : Boolean
#Description: Set to true if the initial splash screen in the installer needs to be shown.
#Example : SHOW_SPLASH_SCREEN = true
#-----
SHOW_SPLASH_SCREEN=true
#-----
#Name      : SHOW_WELCOME_PAGE
#Datatype  : Boolean
#Description: Set to true if the Welcome page in the installer needs to be shown.
```

```
#Example : SHOW_WELCOME_PAGE = false
#-----
SHOW_WELCOME_PAGE=false
#-----
#Name      : SHOW_COMPONENT_LOCATIONS_PAGE
#Datatype  : Boolean
#Description: Set to true if the component locations page in the installer needs to be shown.
#This page only appears if there are products whose installed directory can be changed.
#If you set this to false you will prevent the user from being able to specify alternate
directories.
#Example : SHOW_COMPONENT_LOCATIONS_PAGE = false
#-----
SHOW_COMPONENT_LOCATIONS_PAGE=false
#-----
#Name      : SHOW_CUSTOM_TREE_PAGE
#Datatype  : Boolean
#Description: Set to true if the custom tree page in the installer needs to be shown.
#In this page dependencies can be selected or unselected. This page appears only in a custom
install type.
#Example : SHOW_CUSTOM_TREE_PAGE = false
#-----
SHOW_CUSTOM_TREE_PAGE=false
#-----
#Name      : SHOW_SUMMARY_PAGE
#Datatype  : Boolean
#Description: Set to true if the summary page in the installer needs to be shown.
#The summary page shows the list of components that will be installed in this session.
#Example : SHOW_SUMMARY_PAGE = true
#-----
SHOW_SUMMARY_PAGE=true
#-----
#Name      : SHOW_INSTALL_PROGRESS_PAGE
#Datatype  : Boolean
#Description: Set to true if the install progress page in the installer needs to be shown.
#This page shows the current status in the installation. The current status includes which
product is being installed, which file is being copied.
#Example : SHOW_INSTALL_PROGRESS_PAGE = true
#-----
SHOW_INSTALL_PROGRESS_PAGE=true
#-----
#Name      : SHOW_REQUIRED_CONFIG_TOOL_PAGE
#Datatype  : Boolean
#Description: Set to true if the required config tools page in the installer needs to be
shown.
#This page shows the list of required configuration tools that are part of this installation.
#It shows the status of each tool, including any failures with detailed information on why
the tool has failed.
#Example : SHOW_REQUIRED_CONFIG_TOOL_PAGE = true
#-----
SHOW_REQUIRED_CONFIG_TOOL_PAGE=true
#-----
#Name      : SHOW_OPTIONAL_CONFIG_TOOL_PAGE
#Datatype  : Boolean
#Description: Set to true if the optional config tools page in the installer needs to be
shown.
#This page shows the list of optional configuration tools that are part of this installation
and are configured to launch automatically.
#It shows the status of each tool, including any failures with detailed information on why
the tool has failed.
#Example : SHOW_OPTIONAL_CONFIG_TOOL_PAGE = true
#-----
SHOW_OPTIONAL_CONFIG_TOOL_PAGE=true
#-----
#Name      : SHOW_RELEASE_NOTES
#Datatype  : Boolean
#Description: Set to true if the release notes of this installation need to be shown at the
```

```
end of installation.
#This dialog is launchable from the End of Installation page and shows the list of release
notes available for the products just installed.
# This also requires the variable SHOW_END_SESSION_PAGE variable to be set to true.
#Example : SHOW_RELEASE_NOTES = true
#-----
SHOW_RELEASE_NOTES=true
#-----
#Name      : SHOW_ROOTSH_CONFIRMATION
#Datatype  : Boolean
#Description: Set to true if the Confirmation dialog asking to run the root.sh script in the
installer needs to be shown.
#Valid only in Unix platforms.
#Example : SHOW_ROOTSH_CONFIRMATION = true
#-----
SHOW_ROOTSH_CONFIRMATION=true
#-----
#Name      : SHOW_END_SESSION_PAGE
#Datatype  : Boolean
#Description: Set to true if the end of session page in the installer needs to be shown.
#This page shows if the installation is successful or not.
#Example : SHOW_END_SESSION_PAGE = true
#-----
SHOW_END_SESSION_PAGE=true
#-----
#Name      : SHOW_EXIT_CONFIRMATION
#Datatype  : Boolean
#Description: Set to true if the confirmation when exiting the installer needs to be shown.
#Example : SHOW_EXIT_CONFIRMATION = true
#-----
SHOW_EXIT_CONFIRMATION=true
#-----
#Name      : NEXT_SESSION
#Datatype  : Boolean
#Description: Set to true to allow users to go back to the File Locations page for another
installation. This flag also needs to be set to true in order to process another response
file (see NEXT_SESSION_RESPONSE).
#Example : NEXT_SESSION = true
#-----
NEXT_SESSION=true
#-----
#Name      : NEXT_SESSION_ON_FAIL
#Datatype  : Boolean
#Description: Set to true to allow users to invoke another session even if current install
session has failed. This flag is only relevant if NEXT_SESSION is set to true.
#Example : NEXT_SESSION_ON_FAIL = true
#-----
NEXT_SESSION_ON_FAIL=true
#-----
#Name      : SHOW_DEINSTALL_CONFIRMATION
#Datatype  : Boolean
#Description: Set to true if deinstall confirmation is needed during a deinstall session.
#Example : SHOW_DEINSTALL_CONFIRMATION = true
#-----
SHOW_DEINSTALL_CONFIRMATION=true
#-----
#Name      : SHOW_DEINSTALL_PROGRESS
#Datatype  : Boolean
#Description: Set to true if deinstall progress is needed during a deinstall session.
#Example : SHOW_DEINSTALL_PROGRESS = true
#-----
SHOW_DEINSTALL_PROGRESS=true
#-----
#Name      : component_languages
#Datatype  : StringList
#Description: Languages in which the components will be installed.
```

```

#The following choices are available. The value should contain only one of these choices.
#The choices are of the form Internal Name : External name. Please use the internal name
while specifying the value.
#   en,   : English
#   fr,   : French
#   ar,   : Arabic
#   bn,   : Bengali
#   pt_BR, : Brazilian Portuguese
#   bg,   : Bulgarian
#   fr_CA, : Canadian French
#   ca,   : Catalan
#   hr,   : Croatian
#   cs,   : Czech
#   da,   : Danish
#   nl,   : Dutch
#   ar_EG, : Egyptian
#   en_GB, : English (United Kingdom)
#   et,   : Estonian
#   fi,   : Finnish
#   de,   : German
#   el,   : Greek
#   iw,   : Hebrew
#   hu,   : Hungarian
#   is,   : Icelandic
#   in,   : Indonesian
#   it,   : Italian
#   ja,   : Japanese
#   ko,   : Korean
#   es,   : Latin American Spanish
#   lv,   : Latvian
#   lt,   : Lithuanian
#   ms,   : Malay
#   es_MX, : Mexican Spanish
#   no,   : Norwegian
#   pl,   : Polish
#   pt,   : Portuguese
#   ro,   : Romanian
#   ru,   : Russian
#   zh_CN, : Simplified Chinese
#   sk,   : Slovak
#   sl,   : Slovenian
#   es_ES, : Spanish
#   sv,   : Swedish
#   th,   : Thai
#   zh_TW, : Traditional Chinese
#   tr,   : Turkish
#   uk,   : Ukrainian
#   vi,   : Vietnamese
#Example : COMPONENT_LANGUAGES = {"en"}
#Component : oracle.swd
#-----
component_languages={"en"}
#-----
#Name       : install_type
#Datatype  : String
#Description: Installation type of the component.
#The following choices are available. The value should contain only one of these choices.
#The choices are of the form Internal Name : External name. Please use the internal name
while specifying the value.
#   Complete, : Complete
#   Typical,  : Minimum
#   Custom,   : Custom
#Example : INSTALL_TYPE = "Complete"
#Component : oracle.swd
#-----
install_type="Typical"

```

```

#-----
#Name      : prod_home
#Datatype  : String
#Description: Complete path where the product needs to be installed.
#Example   : PROD_HOME = "C:\ProductName"
#Component : oracle.swd
#-----
oracle.swd:prod_home=<Value Unspecified>
#-----
#Name      : prod_home
#Datatype  : String
#Description: Complete path where the product needs to be installed.
#Example   : PROD_HOME = "C:\ProductName"
#Component : oracle.swd.oui
#-----
oracle.swd.oui:prod_home=<Value Unspecified>
#-----
#Name      : prod_home
#Datatype  : String
#Description: Complete path where the product needs to be installed.
#Example   : PROD_HOME = "C:\ProductName"
#Component : oracle.swd.oui.core
#-----
oracle.swd.oui.core:prod_home=<Value Unspecified>
#-----
#Name      : prod_home
#Datatype  : String
#Description: Complete path where the product needs to be installed.
#Example   : PROD_HOME = "C:\ProductName"
#Component : oracle.swd.jre
#-----
oracle.swd.jre:1.3.1.0.0a:prod_home=<Value Unspecified>
#-----
#Name      : prod_home
#Datatype  : String
#Description: Complete path where the product needs to be installed.
#Example   : PROD_HOME = "C:\ProductName"
#Component : oracle.swd.jre
#-----
oracle.swd.jre:1.3.1.1.0a:prod_home=<Value Unspecified>
#-----
#Name      : prod_home
#Datatype  : String
#Description: Complete path where the product needs to be installed.
#Example   : PROD_HOME = "C:\ProductName"
#Component : oracle.swd.jre
#-----
oracle.swd.jre:1.3.0.0.0:prod_home=<Value Unspecified>
#-----
#Name      : dependency_list
#Datatype  : StringList
#Description: List of Dependees that needs to be installed along with this product.
#The following choices are available. The value can contain any combination of these choices.
#The choices are of the form Internal Name, Version : External name. Please use the internal
name and version while specifying the value.
#   oracle.swd.jre, 1.3.1.1.0a : Java Runtime Environment 1.3.1.1.0a
#   oracle.swd.jre, 1.3.1.0.0a : Java Runtime Environment 1.3.1.0.0a
#   oracle.swd.osp, 2.2.1.2.0 : Oracle Software Packager 2.2.1.2.0 Alpha
#   oracle.swd.oil, 2.2.1.2.0 : Oracle Installation Libraries 2.2.1.2.0
#   oracle.swd.oui, 2.2.1.2.0 : Oracle Universal Installer 2.2.1.2.0 Alpha
#Example   : DEPENDENCY_LIST = {"oracle.swd.jre:1.3.1.1.0a"}
#Component : oracle.swd
#-----
dependency_list=<Value Unspecified>

```

Sample ORAPARAM.INI File

The oraparam.ini file is OUI's initialization file. It should be located in the same directory as the OUI executable file (setup.exe or runInstaller.sh).

```
[Oracle]
DISTRIBUTION=TRUE
SOURCE=../../stage/products.jar
LICENSE_LOCATION=
JRE_LOCATION=../../stage/Components/oracle.swd.jre/1.4.1.3.0a/1/DataFiles/Expanded
JRE_MEMORY_OPTIONS=" -mx96m"
DEFAULT_HOME_LOCATION=
DEFAULT_HOME_NAME=OUIHome
NO_BROWSE=/net
NLS_ENABLED=TRUE
BOOTSTRAP=TRUE
OUI_VERSION=2.3.0.7.0
#SHOW_HOSTNAME=ALWAYS_SHOW shows the hostname panel always
#SHOW_HOSTNAME=NEVER_SHOW does not the hostname panel
#SHOW_HOSTNAME=CONDITION_SHOW shows the hostname panel on condition
SHOW_HOSTNAME=NEVER_SHOW
#THIN_JDBC_FILENAME is optional and defaults to classes12.jar
#The value specified for this should be packaged with OUI, and should
#be relative to <OUI expanded stagedir>/jlib/
THIN_JDBC_FILENAME=classes12.jar
#RUN_OUICA specifies the batch script name that needs to be run
#The script is ouica.bat for win32, and ouica.sh for solaris.
#If the value is not specified, then the OUICA script is not run
RUN_OUICA=ouica.bat

[Certified Versions]
#You can customise error message shown for failure, provide value for CERTIFIED_
VERSION_FAILURE_MESSAGE
Windows=4.0,5.0,5.1,5.2

#Windows NT 4.0
[Windows-4.0-required]
#Service pack for Windows NT4.0, use %SystemRoot%\WinNt\System32\winver.exe to
find version and service pack
SERVICE_PACK=6a
#You can customise the message shown for Service Pack failure through SERVICE_
PACK_FAILURE_MESSAGE
#Minimum display colours for OUI to run
MIN_DISPLAY_COLORS=256
#Use MIN_DISPLAY_COLORS_FAILURE_MESSAGE to customise message for failure of check
for Display colors

[Windows-4.0-optional]
#Minimum CPU speed required for OUI in MHz
CPU=300
#CPU_FAILURE_MESSAGE, use to customise error message for CPU speed check

#Windows 2000 use winver.exe to find version
[Windows-5.0-required]
#Minimum display colours for OUI to run
MIN_DISPLAY_COLORS=256

[Windows-5.0-optional]
#Minimum CPU speed required for OUI
CPU=300
```

```
#Windows XP use winver.exe to find version
[Windows-5.1-required]
#Minimum display colours for OUI to run
MIN_DISPLAY_COLORS=2560
SWAP_SPACE=122

[Windows-5.1-optional]
#Minimum CPU speed required for OUI
CPU=300

#Windows .net use winver.exe to find version
[Windows-5.2-required]
#Minimum display colours for OUI to run
MIN_DISPLAY_COLORS=256

[Windows-5.2-optional]
#Minimum CPU speed required for OUI
CPU=300
```

Index

A

ACCEPT_LICENSE_AGREEMENT, 4-10
-addNode flag, 5-4
addNode script, 5-3

C

cloning considerations, 2-11
cluster
 adding nodes, 5-3
 availability checking, 5-2
 deinstallation, 5-3
 detection, 5-2
 installation, 5-2
cluster installation
 command line options, 5-4
cluster nodes
 adding, 5-3
CLUSTER_NODES, 4-9
Clusterware, installing, 5-1
command line arguments, 2-10

D

debugging mechanisms, A-1
DEINSTALL_LIST, 4-8
deinstallation
 dependees, 2-8
 silent, 2-8
 top level products, 2-8
deinstalling from a cluster, 5-3
Deinstalling Oracle Products, 2-7
DEPENDENCY_LIST, 4-9
Disk Space Requirements, 1-2

E

errors, A-2
 exceptions from action libraries, A-3
 incomplete stage and missing files in file groups, A-3
 out of "temp" space, A-3
 summary of, A-2
 UNCAUGHT_RUNTIME exception, A-3
Exceed, A-3
exit codes, 2-11

F

FROM_LOCATION, 4-6
FROM_LOCATION_CD_LABEL, 4-6

H

help, online, 2-2
Home Selector, 3-4

I

INCLUDE, 4-5
INSTALL_TYPE, 4-10
installation
 from the Web, 2-5
 on a cluster, 5-2
 using a response file, 4-10
installation media, 2-4
installation modes, 2-3
 interactive, 2-3
 silent, 2-4
 suppressed, 2-3
Installing Oracle Products, 2-2
inventory, about, 3-3

J

Java Runtime Environment, 1-2
JRE, 1-2

K

Key Features, 1-1

L

language
 selections, 6-1
LOCATION_FOR_DISK2, 4-6
log files, 2-12

M

Memory Requirements, 1-2
modes of installation, 2-3
 interactive, 2-3

- silent, 2-4
- suppressed, 2-3
- multi-CD support, 2-4, 4-6
- Multiple Oracle Homes (MOH), 3-3

N

- NEXT_SESSION, 4-7
- NEXT_SESSION_ON_FAIL, 4-7
- NEXT_SESSION_RESPONSE, 4-7
- NLS_ENABLED variable, 6-1

O

- online help, 2-2
- Optimal Flexible Architecture
 - characteristics of OFA, 3-6
 - directory structure, 3-6
- OPTIONAL_CONFIG_TOOLS, 4-9
- Oracle homes
 - creating, 3-1
 - default, 3-2
 - directory structure (UNIX), 3-6
 - directory structure (Windows), 3-5
 - introduction, 3-1
 - multiple, 3-3
 - removing, 2-7, 3-2
- Oracle Universal Installer
 - definition, 1-1
- ORACLE_BASE, 3-2
- ORACLE_BASE directory, 3-6
- ORACLE_HOME, 4-7
- ORACLE_HOME directory, 3-6
- ORACLE_HOME_NAME, 4-7
- oraparam.ini file, 2-3
 - sample, B-7
- OUI
 - cloning, 2-11
 - command line arguments, 2-10
 - errors, A-2
 - exit codes, 2-11
 - inventory, 3-3
 - log files, 2-12
 - running, 2-9
 - starting, 2-9
- out of "temp" space error, A-3

P

- pre-requisite checks, 2-1
 - command line parameters, 2-2
- product language selections, 6-1

R

- record mode, 4-3
- Reflection X, A-4
- REMOVE_HOMES, 4-8
- removing Oracle homes, 2-7, 3-2
- requirements
 - disk space, 1-2

- memory, 1-2
- system, 1-2
- response file
 - ACCEPT_LICENSE_AGREEMENT, 4-10
 - CLUSTER_NODES, 4-9
 - creating with record mode, 4-3
 - definition, 4-1
 - DEINSTALL_LIST, 4-8
 - DEPENDENCY_LIST, 4-9
 - format, 4-3
 - FROM_LOCATION, 4-6
 - FROM_LOCATION_CD_LABEL, 4-6
 - general format
 - comments, 4-5
 - headers, 4-5
 - values, 4-4
 - INCLUDE, 4-5
 - INSTALL_TYPE, 4-10
 - location, 4-1
 - LOCATION_FOR_DISK2, 4-6
 - modifying, 4-2
 - NEXT_SESSION, 4-7
 - NEXT_SESSION_ON_FAIL, 4-7
 - NEXT_SESSION_RESPONSE, 4-7
 - OPTIONAL_CONFIG_TOOLS, 4-9
 - ORACLE_HOME, 4-7
 - ORACLE_HOME_NAME, 4-7
 - parameters, 4-5
 - REMOVE_HOMES, 4-8
 - RESPONSEFILE_VERSION, 4-6
 - RESTART_SYSTEM, 4-6
 - sample, B-1
 - SELECTED_LANGUAGES, 4-10
 - SHOW_COMPONENT_LOCATIONS_PAGE, 4-7
 - SHOW_CUSTOM_TREE, 4-7
 - SHOW_DEINSTALL_CONFIRMATION, 4-8
 - SHOW_DEINSTALL_PROGRESS, 4-9
 - SHOW_END_SESSION_PAGE, 4-7
 - SHOW_EXIT_CONFIRMATION, 4-7
 - SHOW_INSTALL_PROGRESS_PAGE, 4-7
 - SHOW_OPTIONAL_CONFIG_TOOL_PAGE, 4-7
 - SHOW_RELEASE_NOTES, 4-8
 - SHOW_ROOTSH_CONFIRMATION, 4-8
 - SHOW_SPLASH_SCREEN, 4-8
 - SHOW_SUMMARY_PAGE, 4-8
 - SHOW_WELCOME_PAGE, 4-8
 - specifying, 4-11
 - TOPLEVEL_COMPONENT, 4-8
 - UNIX_GROUP_NAME, 4-8
 - variable lookup order, 4-4
 - RESPONSEFILE_VERSION, 4-6
 - RESTART_SYSTEM, 4-6
- running OUI after installation, 2-9

S

- SELECTED_LANGUAGES, 4-10
- SHOW_COMPONENT_LOCATIONS_PAGE, 4-7
- SHOW_CUSTOM_TREE_PAGE, 4-7
- SHOW_DEINSTALL_CONFIRMATION, 4-8

SHOW_DEINSTALL_PROGRESS, 4-9
SHOW_END_SESSION_PAGE, 4-7
SHOW_EXIT_CONFIRMATION, 4-7
SHOW_INSTALL_PROGRESS_PAGE, 4-7
SHOW_OPTIONAL_CONFIG_TOOL_PAGE, 4-7
SHOW_RELEASE_NOTES, 4-8
SHOW_ROOTSH_CONFIRMATION, 4-8
SHOW_SPLASH_SCREEN, 4-8
SHOW_SUMMARY_PAGE, 4-8
SHOW_WELCOME_PAGE, 4-8
silent installation
 definition, 4-1
 reasons for performing, 4-2
starting OUI, 2-9
System Requirements, 1-2

T

Tarantella, A-4
target home, 3-3
TEMP/TMP directory, 2-5
TOPLEVEL_COMPONENT, 4-8
troubleshooting
 Exceed, A-3
 Reflection X, A-4
 Tarantella, A-4
 tips, A-3

U

UNIX, special instructions, 2-6
UNIX_GROUP_NAME, 4-8

V

variable values, 4-4

W

Web installation, 2-5

